

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

## **TRABAJO FIN DE GRADO**

**Realización de un piloto de arquitectura de integración de CRM  
con ERP basada en servicios Web**

**Luis Andres Martos Cuevas  
Tutor: Luis de Pedro**

**Junio 2020**





# **Realización de un piloto de arquitectura de integración de CRM con ERP basada en servicios Web**

**AUTOR: Luis Andres Martos Cuevas**  
**TUTOR: Luis de Pedro**

**Dpto. Ingeniería Informática**  
**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**Junio 2020**



## Resumen (castellano)

Unas de las tecnologías más demandadas por grandes y medianas empresas hoy día son los sistemas de gestión de clientes también conocidos como *Customer Relationship Management* (CRM) . Estos productos son ideales para gestionar y construir las relaciones de la empresa con los clientes. El CRM cloud por excelencia y el más demandado hoy día es Salesforce. Esta empresa afincada en San Francisco se ha convertido en los últimos años en una de las empresas tecnológicas más importantes del mundo con unos ingresos cercanos a los 16 mil Millones de dólares, y un crecimiento en España de cerca del 35% anual. Con la implantación de Salesforce en las empresas se hace necesario la sincronización y la comunicación de los diferentes sistemas productivos y de gestión previamente existentes.

Los sistemas de gestión empresarial, *Enterprise Resource Planning* (ERP) , tales como SAP o Oracle ERP , son los más extendidos a nivel global. Estas herramientas manejan los procesos centrales para operar dentro de una empresa: finanzas, RR. HH, servicios, compras y otros. Estos sistemas facilitan el tránsito de información a tiempo real entre diferentes departamentos de manera que las empresas pueden tomar decisiones guiadas por datos y les permite gestionar el rendimiento.

En el presente TFG se plantea una solución de sincronización y comunicación entre Salesforce y cualquier sistema ERP, que ofrezca la posibilidad de consumir sus servicios web, mediante automatismos implementados dentro del propio CRM.

## Palabras clave (castellano)

ERP, Salesforce, Trigger, APEX, Servicios WEB, Computación en la nube.

## **Abstract (English)**

Some of the technologies most demanded by large and medium-sized companies today are the CRM cloud (Customer Relationship Management). These products are ideal for managing and building the company's relationships with customers. The most popular and demanded CRM nowadays is Salesforce. This company seated in San Francisco has become in recent years one of the most important technology companies in the world, with revenues close to 16 billion dollars and growth in Spain of approximately 35% per year. With the implementation of Salesforce in companies it is necessary to synchronize and communicate the different production and management systems that existed previously.

The business management systems, ERP (Enterprise Resource Planning), such as SAP or Oracle ERP, are the most widespread worldwide. These tools manage the central processes to operate within a company: finances, HH. RR, services, purchases and others. These systems facilitate the transit of information in real time between different departments so that companies can make decisions guided by the data and allow them to manage performance.

In this bachelor thesis a synchronization and communication solution are proposed between Salesforce and any ERP system, which offers the possibility of consuming their web services, through automations implemented within the CRM itself.

## **Keywords (inglés)**

ERP, Salesforce, Trigger, APEX, WEB services, Cloud computing





## ***Agradecimientos***

*A mi familia, por todos los esfuerzos y sacrificios que han hecho para que yo llegase hasta aquí.*

*A mis abuelos, Paco y Celia, por hacerme mejor persona y por rodearme de gente maravillosa.*

*A mis amigos, Chava, Doca, Gil, Héctor, Selu y Soto, por todas las experiencias que hemos vivido juntos y por las que aún nos quedan por vivir.*

*A Julián, por depositar su confianza en mí y por enseñarme todo lo que se del mundo del CRM.*

*Por último agradecer a la UAM y a la EPS por formarme como profesional y abrirme las puertas del mundo laboral.*



# ÍNDICE DE CONTENIDOS

<b>1 Introducción .....</b>	<b>1</b>
1.1 Motivación.....	1
1.2 Objetivos.....	2
1.3 Organización de la memoria.....	2
<b>2 Estado del arte .....</b>	<b>3</b>
2.1 Introducción.....	3
2.2 ¿Qué es un CRM?.....	3
2.3 Tecnología Cloud .....	3
2.4 El CRM por excelencia: Salesforce.....	4
2.4.1 Historia de un éxito .....	4
2.4.2 Estrategia cloud .....	5
2.4.3 Salesforce Lightning.....	6
2.4.4 Modelo básico de datos estándar .....	6
2.4.5 Coste .....	6
2.5 Los Sistemas ERP.....	7
2.6 Coexistencia entre Salesforce y un Sistema ERP .....	8
2.7 Conclusiones.....	8
<b>3 Diseño y desarrollo.....</b>	<b>9</b>
3.1 Introducción.....	9
3.2 Requisitos .....	9
3.3 Simulación de un Sistema ERP .....	10
3.3.1 Introducción.....	10
3.3.2 Descripción .....	10
3.3.3 Entorno .....	10
3.3.4 Tabla Comparativa Servicios en la nube .....	11
3.3.5 Modelo de Datos ERP .....	12
3.3.6 Almacenamiento de datos.....	13
3.3.7 Servicio web .....	13
3.3.8 Conclusiones.....	17
3.4 Salesforce.....	18
3.4.1 Introducción.....	18
3.4.2 Entorno .....	18
3.4.3 Herramientas de desarrollo .....	19
3.4.4 Modelo de datos Salesforce.....	20
3.5 Desarrollo Integración Salesforce .....	22
3.5.1 Seguridad en la transferencia.....	23
3.5.2 Sincronización empleados .....	24
3.5.3 Sincronización Clientes .....	25
3.5.4 Sincronización Líneas de Oportunidad .....	27
3.5.5 Sincronización Contactos .....	29
3.5.6 Control de errores .....	30
3.6 Conclusiones.....	30
<b>4 Pruebas y resultados .....</b>	<b>31</b>

4.1	Introducción.....	31
4.2	Pruebas Unitarias.....	31
4.3	Dataset .....	31
4.4	Resultados.....	31
4.4.1	Sincronización empleados .....	31
4.4.2	Sincronización de contactos .....	33
4.4.3	Sincronización PEPS y Cuentas .....	35
4.4.4	Log de errores .....	38
4.5	Conclusiones .....	38
<b>5</b>	<b>Conclusiones y trabajo futuro.....</b>	<b>39</b>
5.1	Conclusiones.....	39
5.2	Trabajo futuro .....	39
	<b>Referencias.....</b>	<b>43</b>
	<b>Glosario .....</b>	<b>45</b>
	<b>Anexos .....</b>	<b>I</b>
A	Modelo básico de datos de Salesforce .....	I
B	Coste .....	II
C	Campos Cliente sistema ERP .....	III
D	Campos Empleado sistema ERP .....	IV
E	Campos PEPS sistema ERP.....	V
F	Campos Contactos sistema ERP.....	VII
G	Campos Cuenta Salesforce .....	VIII
H	Campos Contactos Salesforce .....	IX
I	Campos Oportunidad Salesforce .....	X
J	Campos Línea Oportunidad Salesforce .....	XII
K	Campos LOG Salesforce .....	XIV
L	Campos Usuario Salesforce.....	XV
M	Ejemplo de respuesta JSON .....	XVI
N	Ejemplo de respuesta JSON .....	XVI
O	Tipos de nube .....	XVII
P	Servicios Ofrecidos .....	XVIII
Q	Cálculos económicos .....	XIX
R	Mantenibilidad.....	XX

# ÍNDICE DE FIGURAS

FIGURA 1-1:DIAGRAMA DE GANTT .....	1
FIGURA 1-1: ESQUEMA GENERAL DE SINCRONIZACIÓN .....	2
FIGURA 2-1: CUOTA DE MERCADO EN 2017 .....	4
FIGURA 2-2: ESQUEMA HERRAMIENTA ERP .....	7
FIGURA 3-1: ESQUEMA DE SINCRONIZACIÓN .....	9
FIGURA3-2:MODELO DE DATOS SISTEMA ERP .....	12
FIGURA3.4:ESQUEMA GENERAL SINCRONIZACIÓN AUTOMÁTICA.....	22
FIGURA3.5:ESQUEMA GENERAL SEGURIDAD EN SALESFORCE .....	23
FIGURA3.6:ESQUEMA GENERAL SINCRONIZACION EMPLEADOS .....	24
FIGURA3.7:ESQUEMA GENERAL SINCRONIZACION CLIENTE .....	26
FIGURA3.8:ESQUEMA GENERAL MODIFICACIÓN CLIENTE SALESFORCE.....	26
FIGURA3.9:ESQUEMA GENERAL SINCRONIZACION LÍNEAS DE OPORTUNIDAD .....	28
FIGURA3.10:ESQUEMA GENERAL SINCRONIZACION CONTACTOS .....	29
FIGURA4.1:CONSULTA AL SISTEMA ERP SOBRE EMPLEADOS .....	31
FIGURA4.2:USUARIO GENERADO EN SALESFORCE DESDE SISTEMA ERP .....	32
FIGURA4.3:CONSULTA AL SISTEMA ERP SOBRE JEFES DE PROYECTO .....	32
FIGURA4.4:JEFE DE PROYECTO GENERADO EN SALESFORCE .....	33
FIGURA4.5:CONSULTA SOBRE CONTACTOS SISTEMA ERP.....	33
FIGURA4.6:BUSCADOR CONTACTOS SALESFORCE.....	34
FIGURA4.7:RESULTADOS BUSCADOR SALESFORCE .....	34
FIGURA4.8:PRUEBA SOBRE CONTACTO EN SALESFORCE .....	35
FIGURA4.9:PRUEBA SOBRE CLIENTE EN SALESFORCE .....	35
FIGURA4.9:PRUEBA SOBRE OPORTUNIDAD EN SALESFORCE .....	36
FIGURA4.10:PRUEBA SOBRE LÍNEA OPORTUNIDAD EN SALESFORCE.....	36
FIGURA4.11:PRUEBA SOBRE SINCRONIZACION CLIENTE SALESFORCE.....	37

FIGURA4.12:PRUEBA SOBRE SINCRONIZACION CLIENTE SISTEMA ERP.....	37
FIGURA4.13:PRUEBA SOBRE SINCRONIZACION LÍNEA DE OPORTUNIDAD SALESFORCE .....	37
FIGURA4.14:PRUEBA SOBRE SINCRONIZACION LÍNEA DE OPORTUNIDAD SISTEMA ERP .....	38
FIGURA4.14:EJEMPLO DE LOG AUTOMÁTICAMENTE GENERADO EN SALESFORCE .....	38
FIGURA P-1: ESQUEMA DE ARQUITECTURA CLOUD .....	XVIII
FIGURA Q-1: HERRAMIENTA PAGO SALESFORCE -SAP .....	XIX
FIGURA R-1: GRAFICA DE LOGS .....	XX

## ÍNDICE DE TABLAS

TABLA 1 : CAMPOS CLIENTE SISTEMA ERP.....	III
TABLA 2: CAMPOS EMPLEADO SISTEMA ERP .....	IV
TABLA 3: CAMPOS PEPS SISTEMA ERP .....	VI
TABLA 4: CAMPOS CONTACTOS SISTEMA ERP .....	VII
TABLA 5: CAMPOS CUENTA SALESFORCE .....	VIII
TABLA 6: CAMPOS CONTACTO SALESFORCE .....	IX
TABLA 7: CAMPOS OPORTUNIDAD SALESFORCE .....	XI
TABLA 8: CAMPOS LÍNEA OPORTUNIDAD SALESFORCE .....	XIII
TABLA 9: CAMPOS LOG SALESFORCE .....	XIV
TABLA 10: CAMPOS LOG SALESFORCE .....	XV





# 1 Introducción

## 1.1 Motivación

Hoy en día el sector IT es uno de los que más está creciendo en España y en el Mundo. Cada vez se demandan más profesionales que sepan adaptarse a los tiempos y al entorno cambiante, a la vez que aumentan el número de clientes potenciales. Dentro de este sector, la consultoría informática juega un papel muy importante y cada vez más empresas ofrecen este tipo de servicios.

Dentro de la consultoría informática una de las áreas que impulsa gran parte de este crecimiento es el área de CRM. Este tipo de herramientas se centran en focalizar y realizar una estrategia de negocio centrada principalmente en el cliente. Dentro del mundo CRM hay herramientas tipo “On premise”, como Siebel, que gestionan los datos de los clientes de manera local y se ejecutan dentro de los ordenadores de la propia organización o empresa. Este tipo de Software está quedando cada vez más obsoleto debido a que queda bajo responsabilidad de la empresa el manejo de los datos, teniendo que ocuparse de la seguridad, integridad, disponibilidad, escalabilidad, y capacidad de adaptación de estos afectando también de manera económica a la viabilidad del proyecto. Es por ello por lo que muchas de estas herramientas están dejando de ser ofrecidas a los clientes y se está pasando a un nuevo modelo de negocio: El CRM Cloud.

Las herramientas tipo Cloud basan su modelo en ofrecer soluciones CRM desde la nube, ofreciendo servicios de gestión de datos desde internet. Gracias a la tecnología Cloud se han abaratado los costes de las herramientas y han permitido que las empresas puedan ser más competitivas gracias a la escalabilidad y la gran capacidad de adaptación de este tipo de tecnología, permitiendo así llegar cada vez a más clientes. Es por ello por lo que se viene dando desde hace relativamente poco tiempo una gran transición desde herramientas “On Premise” a herramientas tipo Cloud.

La herramienta por excelencia de tipo CRM Cloud y la más demandada hoy día es Salesforce. Esta herramienta relativamente joven se está adaptando continuamente a las necesidades de las empresas, pero aun así hay ciertas funcionalidades que todavía no están implementadas de manera nativa. Una de estas funcionalidades es la sincronización con los sistemas ERP (Enterprise Resource Planning). Muchos de los clientes llevan la gestión de los recursos de la empresa a través de alguna herramienta ERP, mayoritariamente se utiliza una herramienta llamada SAP.

Las empresas almacenan en sus sistemas ERP la información relacionada con sus proyectos clientes, y RRHH. Estos datos contienen información que comprende el ciclo de vida de un proyecto desde la identificación de este hasta su finalización y facturación. Por este motivo es necesario una total sincronización entre ambos sistemas para evitar incongruencias que puedan afectar gravemente a los recursos de la empresa.

En consecuencia, se demandan soluciones de sincronización que sean eficaces, mantenibles y económicas y en este TFG se mostrará una solución que cumple los tres requisitos. El diagrama de Gantt del proyecto es el siguiente:

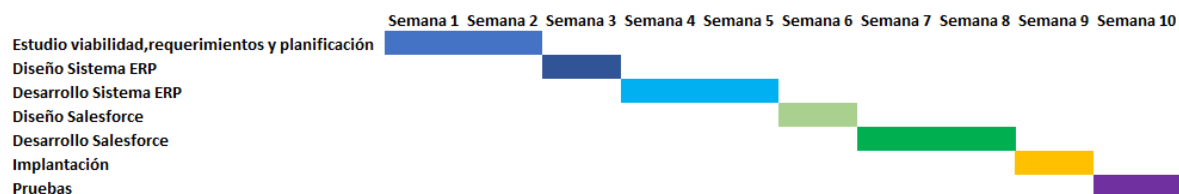
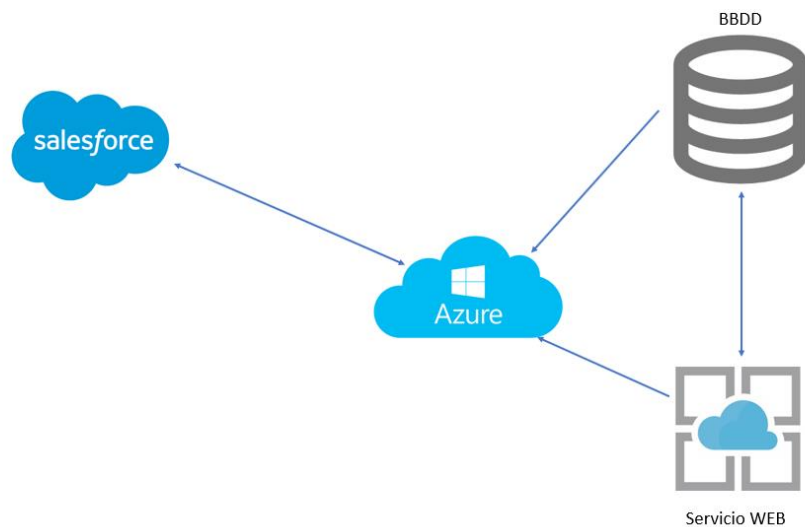


Figura 1-1: Diagrama de Gantt

## 1.2 Objetivos

El objetivo principal de este trabajo es proporcionar una solución sencilla, eficaz y genérica a la problemática de la sincronización entre el CRM Salesforce y cualquier sistema ERP mediante un servicio web publicado por la propia herramienta ERP. Simulando el sistema ERP mediante servicios web alojados en Azure, se construirá un sistema representado en el siguiente esquema:



**Figura 1-1: Esquema general de sincronización**

## 1.3 Organización de la memoria

Este documento consta de cuatro partes principales. La primera de ellas es el estado del arte, en el que se mostraran las tecnologías más punteras dentro de los CRM, concretamente de Salesforce, que están liderando la revolución digital hoy día. En el segundo capítulo, diseño y desarrollo, se explica, desde el punto de vista de ambos sistemas, los entornos utilizados, el modelo de datos y la solución técnica implementada. En la sección de resultados se mostrarán las pruebas realizadas y el resultado de estas. Para finalizar se mostrarán las conclusiones resultantes comentando los puntos de mejora en un futuro.

## 2 Estado del arte

---

### 2.1 Introducción

En este capítulo se dará información sobre el mundo CRM y se abordarán las razones principales por las cuales Salesforce se ha convertido en la plataforma CRM cloud más utilizada en el mundo hoy día, también se describirán las principales características de los sistemas ERP y como puede coexistir Salesforce con estos sistemas.

### 2.2 ¿Qué es un CRM?

En todo proceso comercial el cliente siempre debe ser pilar fundamental sobre el que se erigirán todas las relaciones comerciales. Saber identificar que clientes actuales o potenciales pueden ser más beneficiosos es primordial para adoptar una estrategia comercial que permita el crecimiento de la empresa. Para identificar que clientes son beneficiosos para el negocio es necesario realizar un estudio de los mismo, recopilando información. Los CRM se basan en toda esa información para ofrecer ofertas personalizadas, conocer cuáles son las necesidades reales de los clientes y otorgar un servicio personalizado que aumente la satisfacción de los servicios prestados. En este es el principio básico sobre el que se asienta la tecnología CRM, concretamente en tres áreas básicas: la gestión comercial, el marketing y el servicio de atención al cliente. Esta estrategia comercial no es nueva, de hecho, es tan vieja como el propio comercio. Un ejemplo sencillo de CRM rudimentario es el de un tendero. El tendero recomendará a sus clientes productos basados en sus adquisiciones previas, cuidará y tratará mejor a aquellos clientes que le reporten un mayor beneficio. Debido a que el volumen de un tendero de barrio no es muy grande, este podría memorizar toda la información relacionada con los clientes, pero esta solución no es válida para empresas con un gran volumen de negocio, es en este punto donde entra en juego el software CRM.

El software CRM engloba todas las herramientas informáticas que permiten administrar, planificar y automatizar todos los procedimientos y tareas necesarias para una correcta relación de una empresa con sus clientes. Los primeros software CRM surgieron en la década de los 90's siendo uno de los pioneros "Siebel CRM Systems Inc". Esta compañía fue fundada por Thomas Siebel y Patricia House en 1993 y fue el número en comercialización de software CRM en 2002. Posteriormente fue adquirida por Oracle en 2006 por unos 5.8 billones de dólares.

La herramienta Siebel CRM ofrece soluciones para la gestión de ventas, marketing, integración de datos de clientes y multitud de soluciones enfocadas al cliente y la gestión empresarial. Es un software que en comúnmente se denomina *On Premise*, es decir, que gestiona los datos de manera local y queda bajo responsabilidad de la empresa la integridad de la información. Pese a ser una herramienta antigua y poco amigable para el usuario según los estándares actuales, se sigue utilizando hoy día en multitud de empresas y ámbitos, aunque cada vez más está siendo remplazada por nuevos CRM con tecnología Cloud. Debido a esto Oracle introdujo una nueva versión de Siebel que corría en la nube y ha aumentado el soporte de la herramienta hasta al menos el 2030.

### 2.3 Tecnología Cloud

La tecnología *Cloud*, también llamada computación en la nube, permite la posibilidad de ofrecer servicios de computación a través de internet. Fueron las grandes empresas como Amazon, Google las que comenzaron ofreciendo estos servicios creando estructuras de computación distribuidas de manera horizontal, con una gran capacidad de escalado y funcionamiento prácticamente ininterrumpido. Esto permitía ofrecer los servicios haciendo que el usuario pagara únicamente por la capacidad de cómputo utilizada en lugar de pagar una máquina independientemente del uso que se hiciera de esta.

Gracias a la computación en la nube el número de servicios basados en red ha aumentado drásticamente en los últimos años. Esto genera una gran cantidad de beneficios tanto para los proveedores, ya que los pueden ofrecer de forma más rápida y eficiente, como para los usuarios, que tienen la posibilidad de acceder a ellos de manera inmediata y de forma transparente. Esto también le ahorra al usuario los costes salariales o los costes de inversión económica (maquinas, locales, redes...) lo que hace que este tipo de tecnología haya aumentado en los últimos años y augurando un gran futuro. Según el IDC *International Data Corporation*, en 2020 cerca del 60% del gasto en TI estará relacionado con la nube y representará entre en 60 y el 70% de todos los gastos de software, servicios y tecnología. Puede consultar los diferentes tipos de nube y los servicios que esta ofrece en los siguientes anexos:

- Tipos de nube: [Tipos de nube](#)
- Servicios ofrecidos por la nube: [Servicios Ofrecidos](#)

## 2.4 El CRM por excelencia: Salesforce

En el siguiente apartado se hablará de como Salesforce se ha convertido en el CRM nº 1 en el mundo, las estrategias y características que lo han llevado hasta allí, así como unas pequeñas pinceladas sobre sus costes y su modelo de datos estándar.

### 2.4.1 Historia de un éxito

Salesforce es un CRM de tipo cloud, concretamente se basa en el paradigma Software como servicio *Software as a Service* (SaaS), y se ha convertido en el CRM líder en todo el mundo. Salesforce nació en 1999 fundada por el exdirectivo de Oracle Marc Benioff, en un tiempo en el que poco se hablaba de la computación en la nube. Desde sus inicios tuvo que competir con otras herramientas existentes como Siebel o Sugar CRM, pero actualmente ha dejado a todos sus competidores atrás obteniendo una cuota de mercado cercana al 20%, muy lejos queda su competidor más cercano, Oracle, con una cuota del 7%. Nominada como una de las empresas más innovadoras del mundo, en los últimos seis años, según Forbes, ha sido, asimismo, la primera empresa de cloud en alcanzar los diez billones de dólares de facturación a nivel mundial, contando con cerca de 250.000 usuarios en 2018.

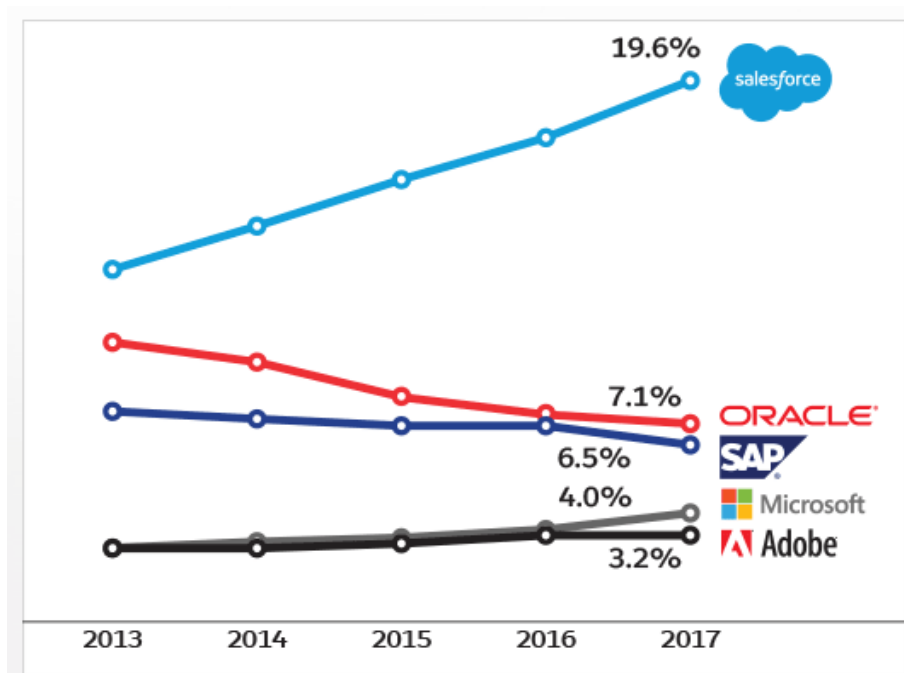


Figura 2-1: Cuota de mercado en 2017

Los 4 pilares sobre los que se fundamenta su éxito son:

- La relación con el cliente: El crecimiento del cliente repercute directamente en el crecimiento del propio CRM, por lo que Salesforce mantiene una estrecha relación con el cliente, ofreciendo servicios personalizados, así como constantes actualizaciones de la plataforma que mejoran la seguridad, arreglan fallos, e introducen nuevas mejoras a petición de los propios usuarios.
- Innovación: Salesforce organiza cada año en California unos de los eventos tecnológicos más importantes a nivel mundial: El dreamforce. En este evento se presentan las novedades más importantes que se incluirán en el año en curso. Salesforce es una plataforma viva que se renueva constantemente, incorporando funcionalidades importantes como por ejemplo las nuevas capacidades de Einstein, un nuevo asistente virtual basado en inteligencia artificial que permite a todos los usuarios de Salesforce:
  - Descubrir perspectivas que aportan claridad acerca de los clientes de su compañía.
  - Predecir resultados para que sus usuarios puedan tomar decisiones con confianza.
  - Recomendar las mejores acciones para sacar el máximo provecho de cada participación.
  - Automatizar tareas rutinarias de modo que sus usuarios puedan centrarse en el éxito del cliente.
- Filantropía : Salesforce destina parte de sus beneficios y recursos a mejorar el mundo actual, llevando a cabo una política que han llamado “1-1-1”. Esta política tiene como objetivo destinar el 1% del tiempo de sus empleados se invierte en horas de servicios comunitarios, el 1% de su capital en donaciones y el 1% de sus productos a ONGs y educación
- Mobile-first: Esta política se centra en ofrecer los mismos servicios que se ofrecen para aplicaciones de escritorio en versiones para dispositivos móviles, siendo un factor diferencial frente al resto de sus competidores.

Gracias a estos 4 pilares Salesforce se ha ganado la confianza de grandes compañías como: Coca Cola, Amazon, Canon, Philips, T-Mobile, Toyota, American Express, Telefónica, Banco Santander, Spotify y trabaja conjuntamente con ellos para ofrecer un mejor servicio a sus clientes.

## 2.4.2 Estrategia cloud

Salesforce adopta una estrategia cloud peculiar: el multi-tenant. Este tipo de arquitectura se basa en una idea básica sencilla: tener una sola base de código que se ejecuta para todos los clientes, con una misma estructura de datos, pero áreas de datos separadas para cada uno. Se puede hacer una analogía con un bloque de pisos, en el que existen infraestructuras comunes, como el ascensor o el rellano, y un plano idéntico para cada uno de los apartamentos, aunque lógicamente cada uno tiene su propio espacio. Esto mejora significativamente la optimización de sus centros de procesamiento de datos, ya que actualizar la base de código significa actualizarla para todos los clientes a la vez y además siendo todos partícipes de las mejoras de inmediato. Siguiendo nuestra analogía anterior sería como instalar un ascensor más rápido, todos los inquilinos se beneficiarían al instante de esta mejora sin tener que hacer modificaciones dentro de sus propias viviendas.

Pero este modelo cloud también tiene sus desventajas: se debe garantizar el servicio de todos los clientes que comparten este espacio. Debido a esto Salesforce define limitaciones de cómputo para garantizar un reparto equitativo de los recursos. Si un cliente necesita más capacidad de cómputo Salesforce encola sus peticiones y las realizará cuando la demanda a si lo permita, lo que hace que ciertas transacciones como las sincronizaciones masivas o las consultas de sobre grandes volúmenes de datos no se realicen de manera instantánea lo que puede ocasionar falta de feedback a los usuarios y demoras en la ejecución de ciertos procesos.

### 2.4.3 Salesforce Lightning

Salesforce originalmente contaba con una interfaz de usuario y diseño acordes a la época en la que fue creado, a principio de los 2000, por lo que ha ido quedando cada vez más desfasada, esta aproximación inicial recibió el nombre de Salesforce Classic.

Frente a las peticiones de los usuarios para modernizar la interfaz y la experiencia dentro de la plataforma, Salesforce respondió con Salesforce Lightning. Salesforce Lightning incluye una nueva interfaz de usuario llamada Lightning Experience, que rediseña completamente los botones, ventanas, textos y gráficos haciéndolos más amigables y visuales para el usuario. Junto con esta nueva interfaz también se introdujeron mejoras para maximizar la productividad de la herramienta algunas de las cuales son:

- Lightning App Builder, que proporciona capacidades de arrastrar y soltar para facilitar la creación y personalización de la aplicación.
- Lightning Component Framework, que incluye herramientas y extensiones que permiten el desarrollo de componentes reutilizables y aplicaciones independientes y la personalización de la aplicación móvil Salesforce.
- AppExchange: Un mercado de aplicaciones, como pudiera ser Google Play o App Store, en el que se ofrecen aplicaciones de Cloud-Computing compatibles con Salesforce.

Sin duda Salesforce Lightning ha contribuido al rápido crecimiento y expansión de Salesforce.

### 2.4.4 Modelo básico de datos estándar

Para saber el modelo básico de datos de Salesforce y los objetos principales dentro de este, consultar el anexo A [Modelo básico de datos de Salesforce](#)

### 2.4.5 Coste

En el anexo B [Coste](#) se detalla el coste de Salesforce dependiendo de las necesidades de cada empresa.

## 2.5 Los Sistemas ERP

Los sistemas planificación de recursos empresariales (en inglés ERP, Enterprise Resource Planning) son sistema de gestión y automatización de muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa. Desde la logística y finanzas hasta la gestión de los recursos humanos. Estos sistemas están compuestos de diferentes módulos con finalidades diferentes cuyos objetivos principales son:

- Optimización de los procesos empresariales.
- Acceso a la información.
- Posibilidad de compartir información entre todos los componentes de la organización.
- Eliminación de datos y operaciones innecesarias de reingeniería.

Para garantizar la integridad de la información en estos sistemas críticos es necesario una correcta integración con el resto de las herramientas que conviven en la empresa, ya que desde estos sistemas se controla el envío de nóminas, la facturación a los clientes, las altas y bajas del personal, por ejemplo.

Algunos de estas herramientas ERP cuentan con módulos CRM, como SAP, pero no han tenido mucho éxito entre los usuarios, debido a su alto coste comparado con los softwares dedicados exclusivamente al CRM, al tiempo de implantación, y una tecnología poco enfocada a la computación Cloud.



**Figura 2-2: Esquema herramienta ERP**

## **2.6 Coexistencia entre Salesforce y un Sistema ERP**

Debido a que ambos sistemas pueden contener el mismo tipo de información (contactos, clientes, personal...etc) se hace necesaria una integración y sincronización para mantener la integridad de los datos. El problema es que Salesforce no dispone de ningún conector estándar que sea compatible con los ERP del cliente, y por parte de los ERP tampoco existe un conector específico para Salesforce. ¿De qué manera simple y eficaz podemos integrar ambos sistemas? La respuesta son los servicios WEB. La mayoría de los ERP contienen dentro de su funcionalidad la capacidad de publicación de servicios web para la entrada y la salida de datos, y a su vez, Salesforce tiene la capacidad de consumir servicios web entre ellos REST, API o SOAP, por lo que es posible establecer una comunicación bidireccional automatizada entre ambos sistemas que garantizara la integridad de los datos y el correcto flujo de la cadena productiva dentro de la empresa.

## **2.7 Conclusiones**

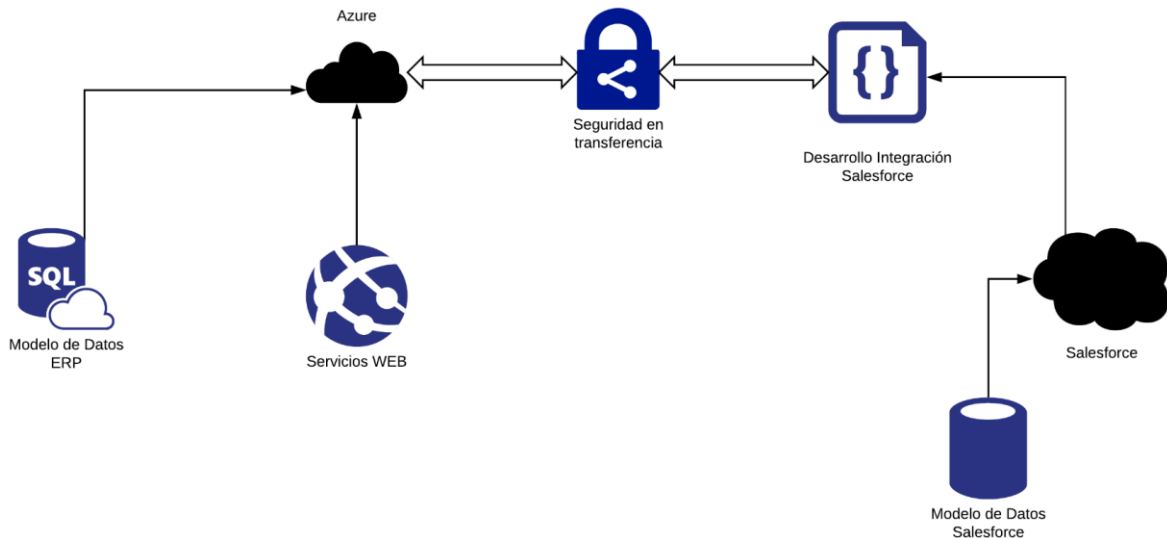
Gracias a las estrategias utilizadas por el equipo de Salesforce, así como las características inherentes a este, no es de extrañar que se haya convertido en la herramienta cloud CRM más utilizada, con una previsión de futuro muy prometedora, que sin duda le hará seguir siendo el rey de las relaciones con el cliente. Esta rápida expansión de su negocio requiere conexiones con otros sistemas ya existentes como los sistemas ERP, que gestionan muchos aspectos no comerciales dentro de la empresa.



## 3 Diseño y desarrollo

### 3.1 Introducción

En este capítulo se mostrarán las soluciones técnicas llevadas a cabo para lograr la sincronización de ambos sistemas, dividiéndose en dos partes, una el desarrollo llevado a cabo en el sistema ERP y por otro lado el desarrollo que ha sido necesario hacer en Salesforce. De manera general podemos dividir nuestro proyecto en módulos diferenciándolos por funcionalidad:



**Figura 3-1: Esquema de sincronización**

\*En azul se representa todo lo que ha sido necesario construir de cero para llevar a cabo este proyecto

Los componentes de este TFG son los que se pueden ver en la figura 3-1:

- Modelo de datos ERP
- Servicios WEB
- Seguridad en transferencia
- Desarrollo integración Salesforce
- Modelo de datos Salesforce

Por un lado, esta Salesforce, con su base de datos y desarrollo de integración. Después Azure, la nube de Microsoft, compuesto por un servicio web y una base de datos interconectados.

En medio de ambos sistemas contamos con un desarrollo para mantener la seguridad en la transferencia de datos integrado en ambos sistemas.

Por regla general el sistema ERP es el maestro dentro de esta sincronización, ya que suele ser el sistema que más tiempo lleva implantado en la empresa y por el que ya se guían la mayoría de los procesos empresariales, por este motivo empezaremos a describir la implementación y funcionalidad de este sistema

### 3.2 Requisitos

Los requisitos principales para este TFG es integrar Salesforce con un sistema ERP, en tiempo real, con un bajo coste y una mantenibilidad elevada. El sistema ERP ha tenido que ser simulado por los motivos expuestos en el siguiente apartado.

## 3.3 Simulación de un Sistema ERP

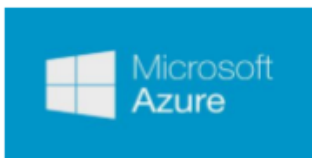
### 3.3.1 Introducción

Hoy en día no hay disponibles licencias gratuitas o software ERP de prueba que permitan la publicación de un servicio web junto con el correspondiente modelo de datos. Es por ello por lo que ha sido necesario simular el funcionamiento de un sistema ERP únicamente centrándose en el servicio web y el modelo de datos, sin añadir funcionalidad adicional que podría estar disponible en las herramientas ERP de pago.

### 3.3.2 Descripción

La simulación del servicio ERP consta de dos componentes principales. Un servicio web desarrollado en C# de tipo API y una base de datos relacional de tipo NoSQL. Ambos sistemas se relacionan entre sí por medio de librerías propias de Microsoft como SQLClient. Estos sistemas se almacenarán y se distribuirán desde Azure.

### 3.3.3 Entorno



El entorno elegido para desplegar y estos dos componentes ha sido Microsoft Azure.

Azure es una nube pública de pago por uso que te permite compilar, implementar y administrar rápidamente aplicaciones en una red global de datacenters (centros de datos) de Microsoft. En el portal existen multitud de soluciones y servicios disponibles para el usuario que le otorga la libertad de crear, administrar e implementar aplicaciones en una red mundial enorme con sus herramientas y marcos favoritos.

Las aplicaciones que se ejecutan en Azure deben funcionar sobre Windows Server 2008 y pueden estar desarrolladas en .NET, PHP, C#, Ruby, Java. El almacenamiento de datos utiliza tablas NoSQL, colas de mensajes o discos NTFS para operaciones de lectura o escritura.

Entre los servicios más destacados de Windows Azure se encuentran aplicaciones Web, máquinas virtuales, bases de datos SQL, back-ends móviles y aplicaciones en remoto.

Los principales motivos por lo que se ha usado esta plataforma en este proyecto fueron:

- **Coste:** Azure es una plataforma de pago por uso, es decir, cuanto más uso hagas de sus servicios más se te facturará a final de mes, sin embargo, Microsoft ofrece la posibilidad a nuevos usuarios de utilizar su plataforma de forma totalmente gratuita durante 12 meses. Este tipo de licencia gratuita tiene ciertas limitaciones, pero en este proyecto no se van a llegar a ellas. En cuanto a la base de datos, con este tipo de cuenta se ofrecen 250GB de almacenamiento, y un número de hasta 10 aplicaciones web por lo que la superación de estos límites es altamente improbable.
- **Facilidad de desarrollo:** Debido a mis conocimientos anteriores del lenguaje C# el desarrollo ha sido sencillo en términos de codificación.

- Herramientas disponibles: Microsoft proporciona un entorno de desarrollo compatible con las librerías y conectores de Azure llamado Visual Studio. Desde este IDE de desarrollo se ha generado desde cero el servicio web. También proporciona otra herramienta para la gestión de la base de datos almacenada en la nube, desde la que se pueden crear las tablas, insertar datos, modificar columnas, establecer relaciones y claves. Esta herramienta se llama Microsoft SQL Server Management Studio
- Facilidad de despliegue: Desde el Visual Studio desplegar un servicio web en la nube pública de Azure es tan sencillo como darle a un botón.
- Soporte: Microsoft cuenta con un soporte dentro de la plataforma que resuelve cualquier duda y problema, además cuenta con una gran comunidad de usuarios, por lo que hay bastantes recursos de aprendizaje disponibles online.

### 3.3.4 Tabla Comparativa Servicios en la nube

En la siguiente tabla se muestra una tabla comparativa entre las opciones más populares de servicios cloud.

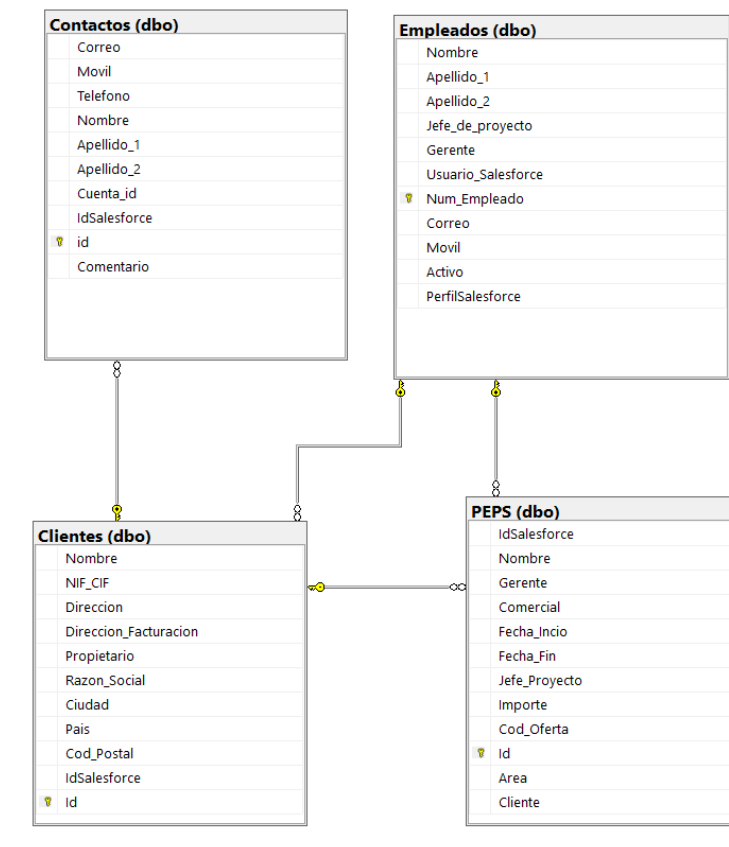
			
<b>Coste</b>	Licencia gratuita limitada durante 12 meses. Coste estimado para este TFG sin licencia gratuita: <b>12\$/Mes*</b>	Licencia gratuita limitada durante 12 meses. Coste estimado para este TFG sin licencia gratuita: <b>4.37\$/Mes*</b>	Licencia gratuita limitada durante 12 meses. Coste estimado para este TFG sin licencia gratuita: <b>4.47\$/Mes*</b>
<b>Facilidad de desarrollo</b>	Conocimientos previos en el lenguaje de desarrollo	Sin conocimientos previos acerca de las herramientas ni lenguaje	Sin conocimientos previos acerca de las herramientas ni lenguaje
<b>Herramientas</b>	Integrado con Visual Studio y Microsoft SQL Management Suite	Extensión para visual Studio. MySQL Client para BBDD.	Integración con visual Studio Code con una extensión.
<b>Despliegue</b>	Un clic desde Visual Studio	Despliegue mediante línea de comando con el SDK de Google.	Despliegue con herramientas como Jenkins
<b>Soporte</b>	Soporte limitado sin previo pago	Soporte limitado sin previo pago	Soporte limitado sin previo pago

\*Precios calculados a día 16/01/2020 para una instancia SQL de 1GB ubicada en EE. UU. y el servidor API más básico ofrecido por cada una de las plataformas.

### 3.3.5 Modelo de Datos ERP

En este capítulo discutiremos el modelo de datos de nuestro sistema ERP , los motivos por los cuales lo hemos elegido y la explicación de las diferentes entidades que lo componen.

El modelo de datos utilizado en el sistema ERP ha sido el siguiente:



**Figura3-2:Modelo de datos sistema ERP**

Se han elegido estas 4 entidades debido a que son las principales que formarían la fuerza de venta de una empresa estándar. Entrando más en detalle en cada una de ellas y sus relaciones, tendríamos. Para la creación de estas entidades se ha utilizado la herramienta Microsoft SQL Server Managment Studio, que a través de su interfaz gráfica permite la creación de tablas, campos, llaves, y relaciones sin necesidad de utilizar código SQL, lo que facilita y hace más trivial la generación de las entidades y sus campos explicados a continuación:

#### 3.3.5.1 Clientes

A quien vendemos nuestros productos o servicios. Las empresas necesitan guardar información de sus clientes ya sea para facturarles, proponer ofertas o simplemente establecer relaciones empresariales con ellos. El eje central de nuestra operativa de venta. En Salesforce (concretamente en Sales Cloud) es el objeto central sobre el que construiremos nuestra plataforma. Este objeto está relacionado con la entidad “Empleados”. Esta relación se basa en el hecho de que un empleado generara una oportunidad o realizara una venta sobre un cliente, a fin de repartir comisiones y ver el rendimiento de cada uno de los comerciales de la empresa en función de las ventas realizadas.

Los campos creados para esta entidad se pueden consultar en el anexo: C [Campos Cliente sistema ERP](#)

### 3.3.5.2 Empleados

Son los empleados de la empresa que darán de alta las oportunidades en Salesforce y trabajarán sobre los clientes a fin de vender nuevos proyectos y generar negocio sobre la empresa.

Este objeto está relacionado con las demás entidades, con **clientes** debido a que los empleados darán de alta oportunidades sobre ellos que luego generaran PEPS que pasaran a nuestro sistema ERP simulado.

Para consultar los campos de los Empleados ir a anexo: D [Campos Empleado sistema ERP](#)

### 3.3.5.3 PEPS

El plan de estructura del proyecto (PEP) es un modelo que representa el trabajo por realizar dentro de un proyecto y que se organiza en forma de estructura jerárquica. El PEP es una pieza fundamental dentro de la herramienta ERP, ya que proporciona una visión de conjunto de todo el proyecto y constituye la base para poder organizarlo y coordinarlo. También muestra el volumen de trabajo, el tiempo requerido y los costes del proyecto. Asimismo, es la base funcional a partir de la que se crean los siguientes pasos de planificación de un proyecto como, por ejemplo, la planificación de procesos, planificación de costes, programación, planificación de fechas y capacidad o cálculo de costes, así como control de proyecto

Para consultar los campos de los PEP ir a anexo: E [Campos PEPS sistema ERP](#)

### 3.3.5.4 Contactos

En esta tabla se almacenará la información relacionada con los contactos. Haría el papel de directorio, donde los empleados podrían consultar por empresa sus contactos asociados. Estos contactos podrán ser importados a Salesforce por medio del servicio de sincronización.

Para consultar los campos de contactos ir al anexo: F [Campos Contactos sistema ERP](#)

## 3.3.6 Almacenamiento de datos

Los datos recopilados por Salesforce y los que se carguen directamente en el sistema, serán almacenados en un base de datos que proporciona el propio Azure de manera gratuita. En este caso debido que no se espera una gran volumetría se ha optado por una instancia SQL gratuita con capacidad de 1GB. Esta base de datos es accedida por el servicio ERP a través de un servidor gratuito, que también proporciona Microsoft, ubicado en EE. UU.

## 3.3.7 Servicio web

Para manejar el flujo de información entre Salesforce y las bases de datos de Azure se ha creado un servicio WEB de cero realizado íntegramente para este proyecto. El servicio se alojará en Azure y forma parte de la licencia gratuita que proporciona Microsoft. En este capítulo se mostrarán las soluciones técnicas llevadas a cabo para realizar este desarrollo.

### 3.3.7.1 Herramientas utilizadas

Se ha utilizado la aplicación de desarrollo de Microsoft visual Studio ya que ofrece la posibilidad de crear un proyecto con la estructura básica de un servicio de API REST, además de proporcionar metadatos para conexión con la base de datos de Azure. También ofrece la posibilidad de desplegar el servicio de manera local para poder realizar pruebas y del mismo modo desplegarlo directamente en la aplicación web de Azure.

### 3.3.7.2 Lenguaje utilizado

El desarrollo se ha realizado utilizando el lenguaje C#. Aunque no se ha estudiado en la carrera se ha estudiado uno prácticamente igual(C) por lo que el desarrollo no ha requerido una curva de aprendizaje muy compleja.

### 3.3.7.3 Tipología

Se ha optado por realizar un servidor de tipo API REST. Los principales motivos por los que se ha elegido esta tipología han sido:

- Estructura básica proporcionada por Visual Studio: Esta herramienta permite la creación de un proyecto API REST básico con los controladores y archivos básicos para su funcionamiento.
- Facilidad de llamada desde Salesforce: Salesforce permite realizar llamadas a servicios web de tipo REST con relativa facilidad utilizando funciones nativas, mientras que, si hubiera utilizado otro tipo de servicio, por ejemplo, SOAP hubiese sido necesario crear varias clases y la complejidad del código se hubiera incrementado significativamente.
- Transformación de datos: Salesforce contiene funciones nativas para el manejo de datos que contengan el formato JSON propio de los API REST, por lo que el tratamiento de los datos que se reciben y envían al servicio lo que ha disminuido la complejidad del desarrollo.

### 3.3.7.4 Estructura

Para poder realizar un mejor desarrollo del servicio y un mejor mantenimiento, se ha optado por definir tres clases por entidad (Clientes, Contactos, Empleados y PEPS) cada una de estas clases tienen una función específica dentro del servicio y se encuentran agrupadas por funcionalidad siguiendo la siguiente estructura de carpetas.

- Models: Dentro de esta carpeta se encuentran las clases que contienen la definición del objeto correspondiente a cada entidad (campos, tipo de campo, getters y setters). Su función es definir y construir el objeto que será retornado a Salesforce o el que llegará desde Salesforce para ser introducido en la base de datos. Hay una clase para cada una de las entidades: "Clientes.cs", "Contactos.cs", "PEPS.cs", "Empleados.cs".
- BBDD: Se encuentran las clases encargadas de la llamada a las bases de datos mediante consultas SQL para cada una de las entidades y poder recoger los datos y realizar las operaciones de inserción, actualización o consulta de los datos. Una vez realizada la consulta se rellenan los objetos contenidos en la carpeta models y se envía la respuesta a Salesforce. Las clases de esta carpeta son: "ClienteHandler.cs", "ContactosHandler.cs", "PEPSHandler" y "EmpleadosHandler.cs".
- Controllers: El controlador web de cada entidad. Recibe la operación a realizar que ha sido invocada por Salesforce y se encarga de redirigirla a su correspondiente método. Realiza las comprobaciones de validación de datos de entrada, y si la respuesta es correcta la devuelve en forma de JSON. Las clases son "clienteController.cs", "ContactosController.cs", "PepsController.cs" y "EmpleadosController.cs".

### 3.3.7.5 Seguridad

Existen varias maneras de securizar un servidor web para su comunicación con Salesforce utilizando por ejemplo certificados o usuario y contraseña. En este caso se ha decidido utilizar seguridad mediante usuario y contraseña debido a que no se considera necesario un nivel de seguridad muy alto para este proyecto. En entornos reales se utilizan llamadas a servicios SOAP autenticados con certificados por parte de Salesforce y del propio servidor.

Para la generación del token de autenticación se han utilizado la misma estructura que la mencionada anteriormente para cada una de las entidades. Una clase "LoginRequest.cs" que contiene la definición del objeto login con los campos usuario y contraseña. Dentro de Controllers una clase "LoginController.cs" que redirige la llamada entrante a un generador de tokens llamando a "TokenGenerator.cs" y en BBDD finalmente tenemos la clase llamada "TokenValidationHandler.cs". El token tiene una validez de 3 horas (configurable) desde que se obtiene, y se utilizan funciones propias de Microsoft para generar un token utilizando el cifrado SHA-256.

La operación para la obtención es: *api/login/authenticate*.

Otro método de seguridad implementado ha sido la utilización de las librerías de Microsoft *System.Data.SqlClient*, que permiten escapar caracteres especiales en los parámetros recibidos que se utilizaran después en las consultas SQL.

### 3.3.7.6 Funcionalidad del servidor

Para cada una de las entidades se han definido una serie de operaciones API. Estas operaciones son llamadas a través de la URL de invocación del servidor. A modo de ejemplo, sí que quiere obtener un cliente por su nombre habría que realizar una llamada del tipo:

*www.---com/api/Cliente/GetByNombre* y adjuntar en el cuerpo de la llamada el nombre por el que se quiere buscar. Estas llamadas desencadenan una conexión con la base de datos y posterior operación SQL.

Las llamadas y consultas SQL contempladas para cada una de las entidades son:

- **Cientes**

- *api/Cliente/GetByNombre*: Devuelve la información asociada a un cliente dado un nombre

- ❖ Consulta: *select \* from Clientes where Nombre LIKE @Name*

- *api/Cliente/GetByNIF*: Devuelve la información asociada a un cliente dado un NIF

- ❖ Consulta: *select \* from Clientes where NIF\_CIF = @NIF*

- *api/Cliente/InsertClient*: Inserta un cliente en la base de datos

- ❖ Consulta: *Insert into Clientes (Nombre,NIF\_CIF,Direccion,Direccion\_Facturacion,Propietario,Razon\_Social,Ciudad,Cod\_Postal,IdSalesforce,Pais) values (@Nombre,@NIF\_CIF,@Direccion,@Direccion\_Facturacion,@Propietario,@Razon\_Social,@Ciudad,@Cod\_Postal,@IdSalesforce,@Pais)*

- *api/Cliente/ModifyClient*: Modifica un cliente en la base de datos dado un NIF como clave.

- ❖ Consulta: *UPDATE Clientes SET Nombre = @Nombre, NIF\_CIF = @NIF\_CIF, Direccion = @Direccion, Direccion\_Facturacion = @Direccion\_Facturacion, Propietario = @Propietario, Razon\_Social = @Razon\_Social, Ciudad = @Ciudad, Cod\_Postal=@Cod\_Postal, Pais=@Pais WHERE (idSalesforce = @idSalesforce AND Id=@Id)*

- **PEPS**

- *api/Ciente/InsertPEP*: Inserta un PEP en la base de datos

- ❖ Consulta: *Insert into PEPS (IdSalesforce,Nombre,Gerente,Comercial,Fecha\_Incio,Fecha\_Fin,jefe\_Proyecto,Importe,Cod\_Oferta,Area,Cliente) values( @IdSalesforce,@Nombre,@Gerente,@Comercial,@Fecha\_Incio,@Fecha\_Fin,@jefe\_Proyecto,@Importe,@Cod\_Oferta,@Area,@Cliente);*

- *api/Ciente/ModifyPEP*: Modifica un PEP en la base de datos dado un id Salesforce como clave.

- ❖ Consulta: *UPDATE PEPS SET IdSalesforce = @IdSalesforce, Nombre = @Nombre, Gerente = @Gerente, Comercial = @Comercial, Fecha\_Incio=@Fecha\_Incio,jefe\_Proyecto = @Jefe\_Proyecto, Fecha\_Fin = @Fecha\_Fin, Importe = @Importe,Cod\_Oferta=@Cod\_Oferta,Area=@Area WHERE (idSalesforce = @idSalesforce AND Id=@Id);*

- **Empleados**

- *api/Empleados/GetEmpleados*: Devuelve todos los empleados contenidos en la base de Datos.

- ❖ Consulta: *select \*from Empleados*

- **Contactos**

- *api/Contactos/GetByName*: Devuelve la información asociada a un contacto dado un nombre

- ❖ Consulta: *select \*from Contactos where Nombre like @Nombre or Apellido\_1 like @Apellido*

- *api/Contactos/GetByCompany*: Devuelve la información asociada a un contacto dado un nombre de compañía

- ❖ Consulta: *select \*from Contactos where Contactos.Cuenta\_id IN (Select Clientes.Id from Clientes where Clientes.Nombre LIKE @compania)*

- *api/Contactos/ InsertContact*: Inserta un contacto en la base de datos

- ❖ Consulta: *Insert into Contactos (Correo, Movil, Telefono,Nombre,Apellido\_1,Apellido\_2,Cuenta\_id,idSalesforce) values( @Correo,@Movil,@Telefono,@Nombre,@Apellido\_1,@Apellido\_2,@Cuenta\_id,@idSalesforce)*



➤ *api/Contactos/ ModifyContact*: Modifica un contacto en la base de datos dado un NIF como clave

❖ Consulta: *UPDATE Contactos SET Correo = @Correo, Movil = @Movil, Telefono = @Telefono, Nombre = @Nombre, Apellido\_1 = @Apellido\_1, Apellido\_2 = @Apellido\_2, Cuenta\_id = @Cuenta\_id WHERE idSalesforce = @idSalesforce;*

Cada una de estas llamadas desencadena una conexión a la base de datos SQL y ejecuta sobre ella las operaciones necesarias de consulta , inserción o actualización. Toda esta lógica se encuentra en la carpeta de “Controllers” dentro del servidor ERP.

Con estas llamadas se devuelve una respuesta de tipo JSON con la información que ha sido requerida o con la confirmación si la operación se ha realizado correctamente, después ese JSON será tratado por Salesforce y ejecutará las acciones necesarias basadas en él.

Todos los JSON devueltos por el servidor tienen una estructura similar para garantizar la homogeneidad del sistema y una correcta escalabilidad en un futuro. Un ejemplo de JSON devuelto por el sistema en la recuperación de datos se puede encontrar en este anexo : M [Ejemplo de respuesta JSON](#): Un ejemplo de JSON devuelto en una operación de inserción o modificación se puede encontrar en este anexo: N [Ejemplo de respuesta JSON](#)

### 3.3.8 Conclusiones

Por todos los motivos expuestos anteriormente(coste, facilidad desarrollo, herramientas , despliegue y soporte) Azure es la plataforma que mejor encaja en nuestros requerimientos y por tanto ha sido la seleccionada para simular nuestro sistema ERP. Gracias a sus características se ha podido realizar una simulación totalmente funcional, no muy distinto a un sistema real, y que permitirá seguir adelante con nuestro desarrollo en Salesforce.

Todo el código del servidor implementado se puede encontrar en el siguiente enlace de GitHub.

<https://github.com/Mrmarthos/TFG/tree/master/ServidorERP>

## 3.4 Salesforce



### 3.4.1 Introducción

Es el pilar central sobre el que gira todo este proyecto. En este apartado se describirán las soluciones técnicas, desarrollos, entornos y funcionalidad que se han realizado dentro de la nube de Salesforce.

### 3.4.2 Entorno

Salesforce proporciona de manera gratuita una plataforma de aprendizaje para desarrolladores llamada “Trailhead”. En esta plataforma se encuentran contenidos y herramientas que permiten a los desarrolladores aprender nuevas habilidades y mejorar las ya existentes. Para poner en práctica las habilidades adquiridas, Salesforce, proporciona SIN coste organizaciones de desarrollo llamadas “Trailhead Playground”.

Un Trailhead Playground es una organización que puede utilizar para completar retos prácticos y probar nuevas funciones y personalizaciones. La única diferencia entre un Trailhead Playground gratuito y una organización de Salesforce real que utilizaría una empresa, es la limitación del número de usuarios que pueden acceder a dicha organización y la limitación espacio de almacenamiento.

Como para este proyecto no era necesario la creación de múltiples usuarios, ni el almacenamiento masivo de datos dentro de Salesforce, este entorno de desarrollo cuadraba perfectamente con las necesidades para realizar este TFG.

El resto de las características (Objetos estándar, funcionalidad, opciones de configuración...etc) son idénticas a una organización utilizada en el mundo real, por lo que no habría problema en aplicar lo desarrollado en la organización de desarrollo en un entorno real de producción.

Los entornos de Salesforce contienen funcionalidad estándar muy limitada. Debido a esto para poder adecuarse a las necesidades de cada cliente es necesario realizar funcionalidad a medida dentro del propio entorno. Haciendo una analogía, las organizaciones Salesforce de inicio son como un “Lego”, te proporcionan las piezas y queda en manos de los desarrolladores y del cliente construir lo que mejor se adapta a sus necesidades y requisitos.

Dentro de estas “piezas” de Lego que proporciona Salesforce podemos distinguir varios grupos.

- **Objetos:** Su función principal dentro de Salesforce es almacenar la información. Dentro de dichos objetos se establecen relaciones con otros objetos, reglas de validación y se definen los campos y su tipología que componen dicho objeto.

Existen dos tipos de objetos en Salesforce:

- **Objetos estándar:** Estos objetos vienen incluidos con Salesforce y tienen una serie de campos listos para ser usados. Dentro de estos objetos también se permite la creación de campos custom. Algunos de estos objetos son “Cuenta”, “Oportunidad” o “Caso”. Mas adelante veremos más en detalle los utilizados para este proyecto.
- **Objetos custom:** Son objetos creados desde cero por el desarrollador. Hay que definir todos los campos y relaciones. Se puede distinguir un objeto custom de uno estándar por el identificador único de cada objeto, ya que los objetos custom acaban siempre por “\_\_c”. Esto también es aplicable para los campos.

Los objetos de Salesforce se corresponderían con una tabla dentro de un sistema relacional SQL.

- **Clases APEX:** Es un lenguaje de secuencia centrado en objetos y muy definido que permite a los desarrolladores ejecutar declaraciones de control de transacciones y flujos en el servidor de la plataforma de Force.com junto con llamadas a la API de la plataforma Lightning. Además, permite añadir lógica comercial a la mayoría de los eventos del sistema. Es muy parecido a Java, por lo que, si se conoce este, la curva de aprendizaje requiere poco tiempo.
- **Componentes lightning:** Son componentes visuales hechos a medida utilizados para representar información, realizar procesos guiados para el usuario o modificar el aspecto general de la interfaz de usuario. Se desarrollan en un lenguaje muy parecido al XML y constan de un controlador para manejar la información o procesos que está escrito en APEX. De manera estándar Salesforce no viene con ninguna clase Apex generada.

### 3.4.3 Herramientas de desarrollo

Para la creación de objetos y campos se ha utilizado la propia interfaz proporcionada por Salesforce dentro de la organización, sin embargo, para el manejo de las clases de APEX, y componentes lightning se ha optado por utilizar una herramienta externa.

Para crear y modificar las clases de Apex y componentes Lightning , Salesforce proporciona una herramienta llamada “Developer Console” que no es más que un editor online de texto(con algunas opciones más) en el que se programa. Esta “Developer Console” es accesible desde el entorno de desarrollo a través de los ajustes. Debido al pobre rendimiento y opciones que ofrece se ha utilizado para este desarrollo el programa gratuito de Microsoft llamado Visual Studio Code.

Descargando una extensión llamada “ForceCode” se puede conectar Visual Studio con la organización de Salesforce que se desee y realizar desde ahí el desarrollo y posterior despliegue.

### 3.4.4 Modelo de datos Salesforce

El modelo de datos utilizado en Salesforce es el siguiente:

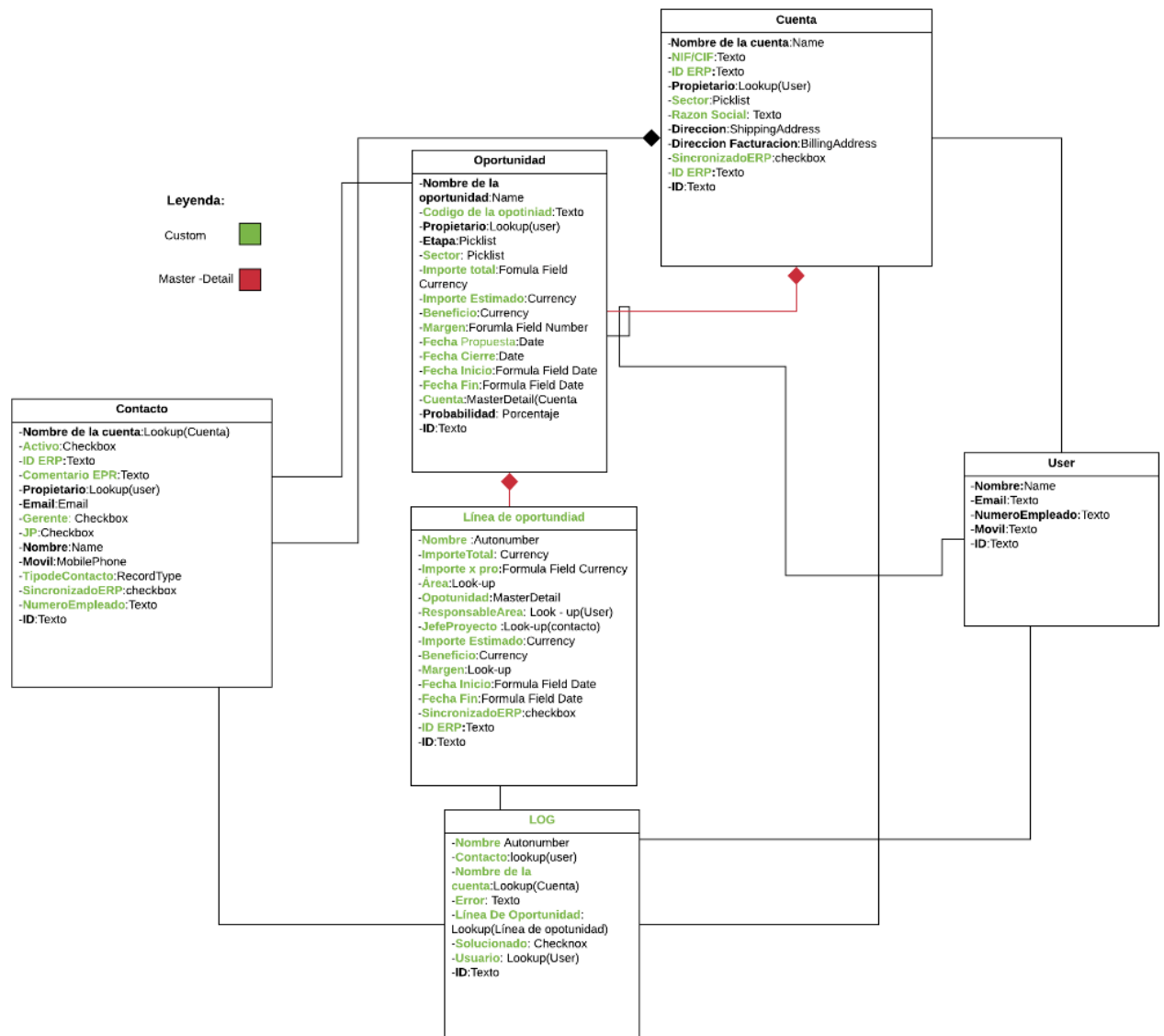


Figura3.2:Modelo de datos Salesforce

En el modelo elegido, el pilar central, es la cuenta. De la cuenta cuelgan las oportunidades con una relación Máster-Detail (si se borra la cuenta se borran las oportunidades hijas), y de las oportunidades las líneas de oportunidad con otra relación Máster-Detail. Este es un modelo bastante estándar en Salesforce ya que por defecto una organización viene con la entidad Cuentas, Contactos, Oportunidades y Usuarios creadas, además de las relaciones entre estos. Estas entidades estándar modificadas han sido las que se han utilizado junto con otras Custom para configurar un modelo de datos que se adecue a la estructura de nuestro sistema ERP y que permita una coexistencia y sincronización entre ambos satisfactoria.

Para generar estas entidades y modificarlas se ha utilizado el gestor de objetos estándar de Salesforce, accesible desde la configuración.

A continuación, se expone una explicación más en profundidad sobre cada entidad:

#### **3.4.4.1 Cuenta:**

Es un objeto estándar dentro de Salesforce que representa a un cliente de la empresa. En él se almacena información tales como su nombre, dirección, CIF...etc. Tiene una correspondencia directa con la entidad “Cliente del sistema ERP”, y formaran parte de nuestra sincronización como veremos más adelante. Sobre esta entidad cliente el empleado generara las oportunidades. Los campos utilizados en esta entidad se pueden ver en el siguiente anexo: G [Campos Cuenta Salesforce](#)

#### **3.4.4.2 Contactos**

Es otro objeto estándar dentro de Salesforce que contiene la información de un contacto relacionado con una cuenta. Sobre él se guarda información relacionada con las personas designadas por el cliente para establecer las relaciones comerciales. Los campos usados en esta entidad se pueden consultar en el siguiente anexo: H [Campos Contactos Salesforce](#)

#### **3.4.4.3 Oportunidades**

Este objeto representa las negociaciones en curso con un cliente. Como en una negociación en la vida real tiene un ciclo de vida representado por etapas. Estas etapas son “Identificada”, “Solicitada”, “Preventa”, “Entregada” “Ganada/Perdida”. Estas etapas son importantes ya que jugaran un papel fundamental en la sincronización con nuestro sistema ERP. Desde este objeto se generarán las líneas de oportunidad que representarán nuestros PEPS. Se pueden crear de 1 a n líneas de proyecto sobre la oportunidad, y servirán para medir el rendimiento comercial y la fuerza de ventas de la empresa. Los campos utilizados en la oportunidad se pueden consultar en el siguiente anexo: I [Campos Oportunidad Salesforce](#)

#### **3.4.4.4 línea de oportunidad**

Este objeto “custom” representa en plan de proyecto de una oportunidad. Tiene una correspondencia con la entidad PEPS de nuestro sistema ERP. Se generará una línea de oportunidad por cada área implicada en la oportunidad o en el proyecto y contendrá información económica, fechas y personas involucradas. Los campos utilizados en esta entidad se pueden consultar en el siguiente anexo: J [Campos Línea Oportunidad Salesforce](#)

#### **3.4.4.5 Log**

Es un objeto custom que almacenara los posibles fallos de sincronización entre ambos sistemas para que puedan ser depurados por un administrador de la plataforma. Los campos usados se pueden consultar en el siguiente anexo: K [Campos LOG Salesforce](#)

#### **3.4.4.6 Usuarios**

Guarda la información relacionada con los usuarios que podrán acceder a la plataforma. Estos usuarios serán los encargados de poblar de información el sistema (Cuentas, oportunidades, Líneas de oportunidad...etc). Los campos se pueden ver en el siguiente anexo: L [Campos Usuario Salesforce](#)

### 3.5 Desarrollo Integración Salesforce

La integración de ambos sistemas se ha realizado mediante llamadas HTTP a los servicios web expuestos de nuestro sistema ERP por parte de Salesforce. Para la realización de estas llamadas se utilizan mecanismos provistos por el propio Salesforce. Podemos distinguir dos tipos de llamadas, aquellas que se realizan de manera automática y las que son realizadas de manera manual por los usuarios.

Para la realización de las llamadas automáticas se han utilizados dos mecanismos proporcionados por Salesforce de manera estándar con la creación de cualquier organización:

- **Triggers:** Los desencadenadores o triggers de Apex permiten realizar acciones personalizadas antes o después de eventos para registros en Salesforce, como inserciones, actualizaciones o eliminaciones. Del mismo modo que los sistemas de bases de datos admiten desencadenadores, Apex permite el uso de desencadenadores para la gestión de registros. Se utilizan para realizar operaciones sobre los registros que cumplan ciertas condiciones, y permiten desencadenar cualquier acción, incluida la llamada a un servicio web de manera asíncrona. Estos Triggers llaman a su vez a las clases de APEX encargadas de la realización de las llamadas HTTP y manejo de datos.
- **Trabajos de Apex:** Permiten programar acciones de Apex en la plataforma. Se utiliza para la realización de procesos que requieran una actualización de los registros de manera regular. Estos procesos se programan mediante los ajustes de administración y llaman a las clases de APEX que implementan una interfaz llamada “Schedulable”. Dentro de esta clase de APEX se implementa toda la lógica, incluidas las llamadas los servicios HTTP de nuestro servidor.

Para la realización de las llamadas manuales por parte del usuario, se han desarrollado componentes lightning únicamente para algunas entidades que permiten las llamadas síncronas a los servicios expuestos por el sistema ERP con parámetros definidos por el usuario. Estas llamadas son síncronas y se explicaran con más detalle más adelante.

El flujo general de la sincronización para las llamadas **automáticas** mediante **triggers** siguen el siguiente diagrama , aplicable para todos los objetos que se sincronizan desde Salesforce:

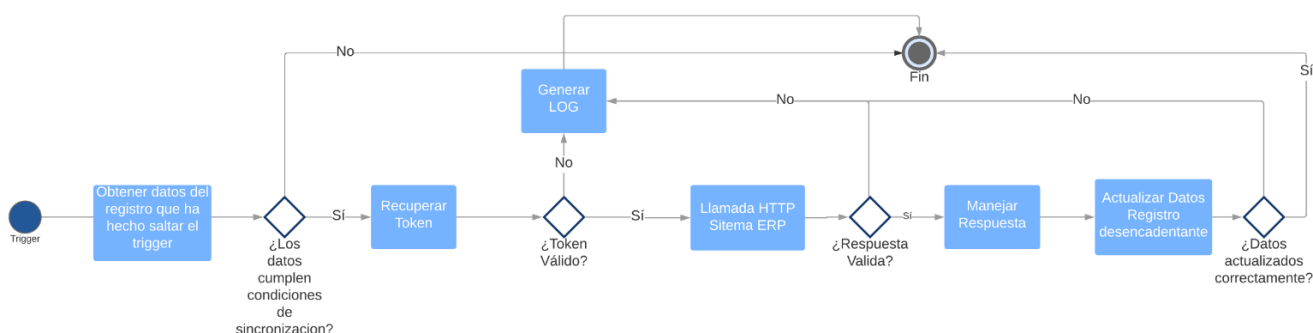


Figura3.4:Esquema general sincronización automática

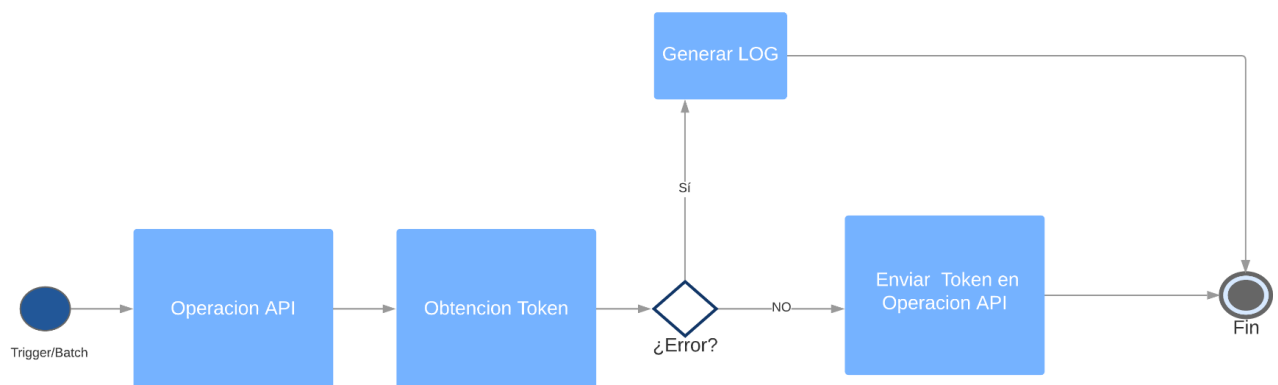
Toda esta lógica está implementada en las diferentes clases de APEX que son llamadas desde los triggers, desde las cuales se realizan las operaciones de consulta, inserción y actualización de datos de la plataforma. Estas operaciones son totalmente transparentes para el usuario y se realizan de manera asíncrona, por lo que no interfieren en la usabilidad de la herramienta.

### 3.5.1 Seguridad en la transferencia

Debido a que se van a sincronizar datos sensibles (Información comercial, datos de empleados...etc) es necesario implementar una política de seguridad acorde con los requisitos. Salesforce ofrece la posibilidad de autenticarse en un sistema externo mediante un certificado auto-firmado generado dentro de la propia organización. Debido a que el proceso de autenticación mediante certificado es complejo de realizar en nuestro servidor ERP y no se considera realmente necesario, se ha optado por una securización mediante usuario y contraseña.

Se ha establecido un usuario y una pass única que autenticara toda la organización de Salesforce en el sistema ERP. Estas credenciales están almacenadas dentro de las organizaciones de Salesforce en un objeto de configuración llamado “Etiquetas Personalizadas”. Esta configuración solo esta visible por un administrador por lo que no es posible que sean modificada ni vista por cualquier otro usuario, asegurando su integridad. Además, esto permite que en caso de actualizar las credenciales solo sea necesario hacerlo en un solo sitio, sin tener que modificar código. Previo paso a realizar cualquier llamada al servicio ERP es necesario realizar una llamada al servicio de autenticación del servidor enviando como parámetros las credenciales almacenadas en Salesforce.

El flujo sería el siguiente:



**Figura3.5:Esquema general seguridad en Salesforce**

### 3.5.2 Sincronización empleados

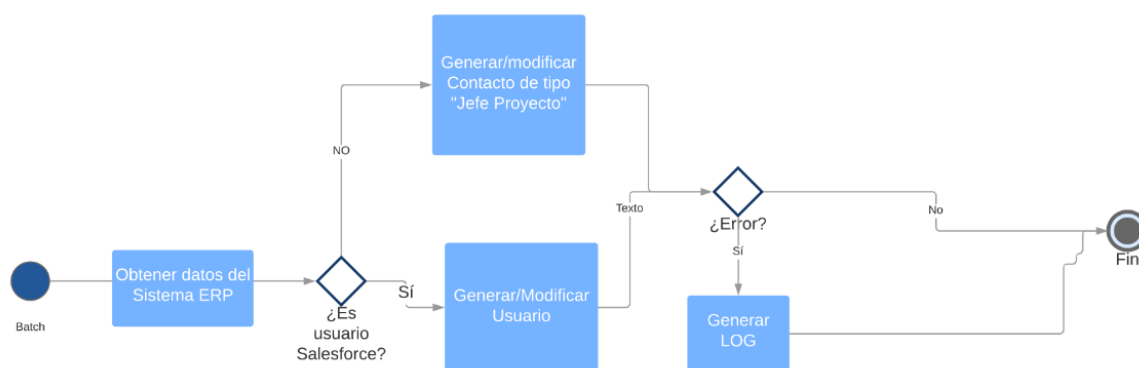
Este servicio se encarga de sincronizar los empleados que se dan de alta en el sistema ERP y creando su correspondiente registro en Salesforce. Esta sincronización es necesaria debido a que hay que mantener ambos sistemas alineados para evitar dar de alta PEPS sobre empleados que ya no están en la empresa o que simplemente hayan pasado a ocupar otros puestos y ya no sea necesario su participación en el proceso de venta. Además, permiten gestionar las licencias de Salesforce de manera correcta para evitar sobrecostos y evitar tener licencias asignadas a personas que ya no lo necesitan.

Esta sincronización se realiza mediante un proceso batch automático que consulta todos los días a determinada hora los empleados del sistema ERP mediante una llamada HTTP a la operación de consulta expuesta por el servidor: *api/Empleados/GetEmpleados*

Una vez obtenida la respuesta se pueden generar y modificar dos tipos de registro dentro de Salesforce:

- **Usuarios:** Si el empleado dado de alta en el sistema ERP tiene el check marcado de “Usuario Salesforce” , se generará un usuario en la plataforma con los datos obtenidos, permitiendo que sea el sistema ERP el que de manera remota administre las licencias de Salesforce, evitando así tener licencias activas ( con el coste que ello conlleva) asignadas a usuarios que ya no la necesitan o que se han dado de baja de la empresa.
- **Contactos:** Si el empleado proveniente del servicio de ERP no tiene marcado el check de “Usuario de Salesforce” se generará un registro especial de Contacto llamado “Jefe de Proyecto” . Este contacto será utilizado más tarde en las sincronizaciones de los PEPS.

El flujo de esta sincronización es el siguiente:



**Figura3.6:Esquema general sincronizacion empleados**

Este servicio se encuentra implementado en las clases APEX llamadas “ActualizadosEmpleados.cls” ,que contiene la lógica de llamada al servicio, y en la clase “ActualizadorEmpleados.cls” que implementa la interfaz para ser llamada desde el proceso Batch.

Con este servicio garantizamos que ambos sistemas se encuentren alineados y que no halla incongruencias en el ámbito de los recursos humanos de la empresa, permitiendo un correcto funcionamiento de ambos sistemas.



Para poblar los registros de **Usuarios** a partir de los datos recibidos desde el sistema ERP se ha establecido el siguiente mapeo entre los campos de Salesforce y los del sistema ERP

Salesforce	ERP
<b>Nombre</b>	<b>Nombre+Apellido1+Apellido2</b>
<b>Email</b>	<b>Correo</b>
<b>Numero Empleado</b>	<b>Num_Empleado</b>
<b>PerfilId</b>	<b>PerfilSalesforce</b>
<b>Gerente</b>	<b>Gerente</b>
<b>Activo</b>	<b>Activo</b>

**Tabla3.1:Tabla mapeo usuario**

Para los **Jefes de Proyecto**:

Salesforce	ERP
<b>Nombre</b>	<b>Nombre+Apellido1+Apellido2</b>
<b>Email</b>	<b>Correo</b>
<b>Numero Empleado</b>	<b>Num_Empleado</b>
<b>IDERP</b>	<b>Id</b>
<b>Movil</b>	<b>Movil</b>

**Tabla3.2:Tabla mapeo jefes de proyecto**

### 3.5.3 Sincronización Clientes

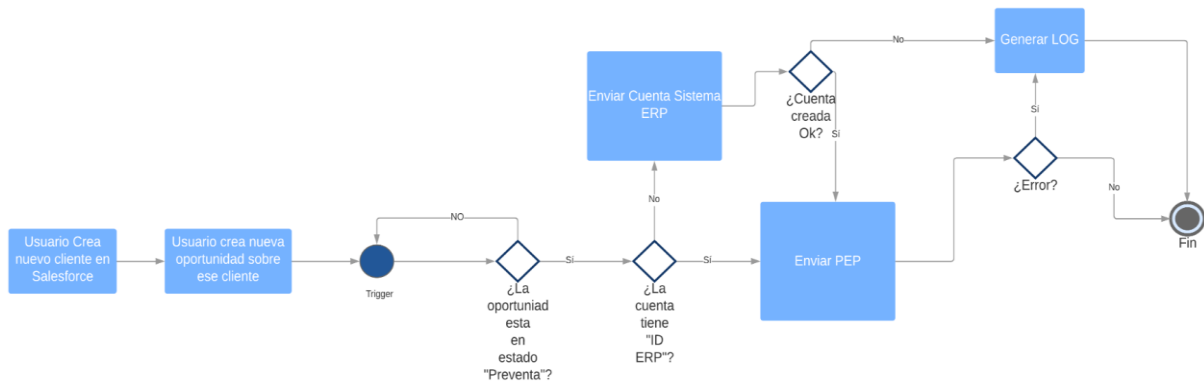
Debido a que los clientes son el pilar central del cualquier CRM, incluido Salesforce, y a que también son una de las partes más importantes de un sistema ERP, es necesario contar con una sincronización activa entre ambas plataformas. El sistema ERP se encargará de la gestión economía de los recursos, y estos tienen que estar asociados correctamente a los clientes para no causar desequilibrios económicos en la empresa. Es por ello por lo que esta sincronización es fundamental que sea bidireccional, es decir que los datos fluyan de Salesforce al sistema ERP y del sistema ERP a Salesforce.

Hay que tener en cuenta que siempre debe haber un maestro de clientes, es decir un sistema en el que los datos siempre sean íntegros y fiables, y otro sistema el cual se nutra de esos datos ya validados. En este caso el maestro de clientes es el sistema ERP, debido a que normalmente este sistema lleva implementado en la empresa más tiempo y su integridad de datos ya ha sido validada, y por otro lado tenemos a Salesforce, la nueva herramienta que vamos a implementar y que se va a nutrir de esos datos previamente validados, con lo cual la posibilidad de incongruencias entre ambas plataformas se ve disminuida.

Previo a la puesta en producción de Salesforce, se debe realizar una carga de datos en la plataforma con los clientes existentes en el sistema ERP. Esto se puede hacer mediante herramientas como Dataloader que cargar los datos de un fichero CSV exportado del sistema ERP.

Una vez cargados los clientes podremos decir que tenemos los dos sistemas completamente alineados con la misma información, y a partir de ahí empezar a trabajar desde Salesforce con la **creación** de nuevos clientes o la **modificación** de los ya existentes.

Para la **creación** de nuevos clientes se sigue el siguiente flujo de sincronización

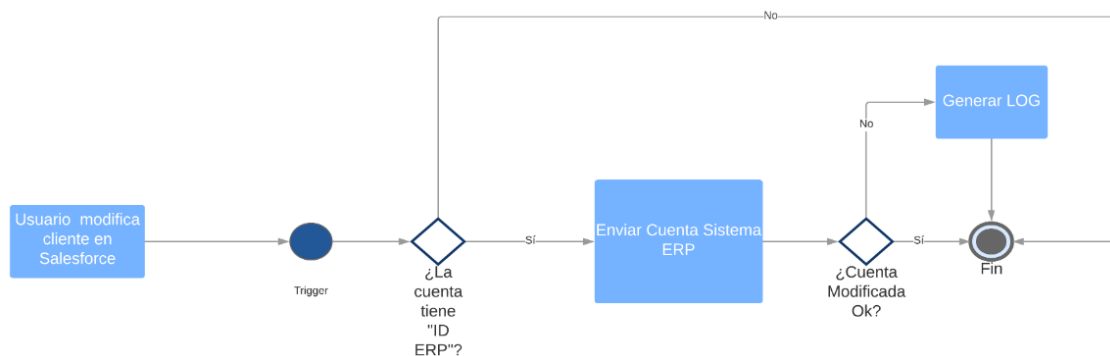


**Figura3.7:Esquema general sincronizacion cliente**

Un usuario generará una nueva cuenta en Salesforce rellenando la información necesaria. En ese momento no es necesaria enviarla al sistema ERP, ya que no hay proyecto sobre ese cliente, simplemente es una “prospección”, con lo cual, a modo de evitar enviar datos “basura” al sistema ERP no se envía de la cuenta justo en el momento de su creación.

Mas tarde, cuando algún usuario genera una oportunidad sobre ese nuevo cliente, y dicha oportunidad para al estado preventa ( en ese estado ya hay gente trabajando y generando gasto en la empresa para formalizar una oferta, por ejemplo) es cuando se envía al sistema ERP. Si la cuenta no tiene informado el “ID ERP” es decir, no ha sido cargada desde ese sistema, se invoca al servicio web de envío de clientes por medio de los triggers y clases necesarias. Si el sistema ERP crea correctamente el cliente, este devolverá el “ID ERP” que se informara en el cliente, quedando ambos sistemas alineados. Con este identificador ya se puede llamar al servicio que crea los PEPS en el sistema ERP sobre ese cliente.

En cuanto a la **modificación** de datos de un cliente el flujo que se utiliza es el siguiente:



**Figura3.8:Esquema general modificación cliente Salesforce**

Para que se produzca esta sincronización tanto de modificación como de inserción se han establecido una serie de reglas de validación dentro del cliente, de tal manera que se rellenen los campos mínimos necesarios que exige el sistema ERP para la creación de un cliente dentro de Salesforce. De esta manera nos aseguramos de que no se generen clientes “no validos” en el sistema ERP ni en Salesforce.

Hay que tener en cuenta que para que esta sincronización funcione y ambos sistemas estén alineados es necesario que los clientes siempre se generen y se modifiquen desde Salesforce. Si esto no se hiciera se podrían machacar los datos previamente creados o modificados en el sistema ERP. De todas formas, se podrían crear los clientes también en el sistema ERP y luego crearlo en Salesforce informando el campo “ID ERP” a mano, pero no se recomienda.

Para el manejo de esta lógica se han creado las siguientes clases y manejadores de triggers en Salesforce:

- **Trigger:** “AccountTrigger.apxt”
- **Manejador Trigger:** “OpportunityHandler.cls”, “AccountHandler.cls”  
“OpportunityUtil.cls”
- **Clases APEX :** “EnvioCuentasERP.cls”, “AccountUtil.cls”, “SendObjectERP.cls”

El mapeo de datos entre el sistema ERP y Salesforce para los clientes es el siguiente:

Salesforce	ERP
Nombre de la cuenta	Nombre
NIF/CIF	NIF_CIF
Direccion	Direccion
Direccion facturación	Direccion facturación
Propietario	Propietario
Razón Social	Razón Social
BillingCity	Ciudad
BillingCountry	Pais
BillingPostalCode	Codigo Postal
IDERP	Id

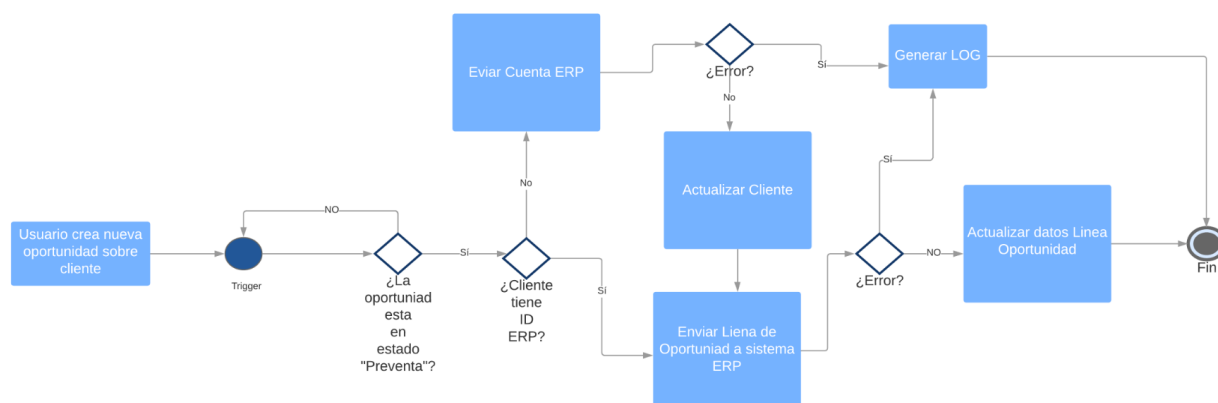
**Tabla3.3:Tabla mapeo clientes**

### 3.5.4 Sincronización Líneas de Oportunidad

Como ya expliqué en apartados anteriores las líneas de oportunidad tiene su correspondencia con los PEPS de nuestro sistema ERP. Las líneas de oportunidad representan el trabajo a realizar dentro de un proyecto para dar respuesta a las necesidades planteadas por un cliente.

La realización de este trabajo por parte de la empresa requiere una planificación previa de los recursos, tanto económicos como humanos. Toda esta información relacionada con el trabajo a realizar se encuentra en las líneas de oportunidad. El gestor de recursos económicos y humanos de la empresa es el sistema ERP por lo que es necesario que los proyectos que se den de alta en Salesforce e incurran en gasto de personal o material se vean reflejados en el sistema ERP para que no ocurran desequilibrios económicos o de personal.

Esta sincronización permite introducir en el sistema ERP estos planes de proyecto de manera automática desde Salesforce y el flujo es el siguiente:



**Figura3.9:Esquema general sincronizacion líneas de oportunidad**

De la misma manera que con los clientes, cuando una línea es sincronizada ,se rellena el “ID ERP” dentro del registro de la línea de oportunidad enviado. De esta manera si se realiza alguna modificación desde Salesforce se envían de nuevo al ERP los nuevos datos utilizando como identificador único ese ID.

El mapeo entre el objeto línea de oportunidad y los PEPS de nuestro sistema ERP es el siguiente:

Salesforce	ERP
<b>id</b>	<b>idSalesforce</b>
<b>Nombre</b>	<b>Nombre</b>
<b>ResponsableArea.NumeroEmpleado</b>	<b>Gerente</b>
<b>Fecha Inicio</b>	<b>Fecha_Inicio</b>
<b>Fecha Fin</b>	<b>Fecha_Fin</b>
<b>JefeProyecto.NumeroEmpleado</b>	<b>Jefe_Proyecto</b>
<b>Importe</b>	<b>Importe</b>
<b>Oportuniad.CodigoOferta</b>	<b>Cod_Oferta</b>
<b>ID ERP</b>	<b>Id</b>
<b>Area</b>	<b>Area</b>
<b>Oportunidad. Cliente</b>	<b>Cliente</b>

**Tabla3.4:Tabla mapeo líneas de oportunidad**

Las clases de APEX que intervienen en esta sincronización son;

- **Trigger:** “OpportunityTrigger.apxt”,” Linea\_de\_oportunidad\_trigger. apxt”
- **Manejador Trigger:** “OpportunityHandler.cls”,” LineadeOportunidadHandler.cls”
- **Clases APEX:** “OpportunityHandler.cls”,” SendObjectERP.cls”

### 3.5.5 Sincronización Contactos

Para la sincronización de los contactos se ha utilizado un enfoque diferente para mostrar algunas de las posibilidades que ofrece Salesforce en cuanto a la integración con distintos sistemas.

Para la obtención de datos desde el sistema ERP se ha implementado un buscador de contactos directamente en la “home” de Salesforce que lanza una consulta sobre la base de datos del sistema ERP mediante una llamada HTTP.

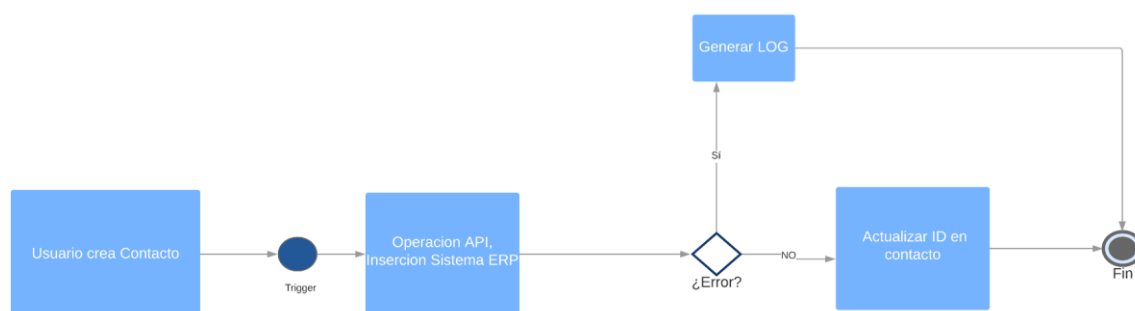
Cuando un usuario de Salesforce en el buscador el nombre de un cliente/compañía, automáticamente lanzara una query sobre el sistema ERP para obtener los contactos relacionados con dicho cliente y poder añadirlos directamente en Salesforce sin tener que realizar cargas masivas y sin la intervención de un Administrador.

Una vez devueltos los datos se ofrece la posibilidad al usuario de añadirlo como un nuevo contacto en Salesforce para más tarde poder operar sobre él.

Para realizar esto se ha desarrollado desde 0 un componente Lightning llamado “buscadorContactos.lex”. Como ya he hablado en capítulos anteriores este componente sigue un modelo MVC donde el MV es el propio componente y el controlador es una clase de APEX llamada “BuscarCorreoController.cls”.

Para enviar los contactos desde Salesforce al sistema ERP se ha seguido el mismo esquema que en sincronizaciones anteriores. Cada vez que se inserta un nuevo contacto se invoca al servicio web que envía los datos al sistema ERP, devolviendo un id , y actualizando esta información en Salesforce.

De manera gráfica podemos resumir el flujo de inserción de la siguiente manera:



**Figura3.10:Esquema general sincronizacion contactos**

El mapeo entre el objeto línea de oportunidad y los PEPS de nuestro sistema ERP es el siguiente:

Salesforce	ERP
id	idSalesforce
Nombre	Nombre
Apellido1	Apellido_1
Apellido2	Apellido_2
AccountID	CuentaID
ID ERP	Id

**Tabla3.4:Tabla mapeo contactos**

### 3.5.6 Control de errores

Debido a que este desarrollo es complejo custom no existe ningún control de errores estándar que sea adecuado para este proyecto. Es por ello por lo que se ha optado por la utilización de un objeto custom llamado LOG.

Para poner en contexto por que se ha tenido que tomar esta decisión de implementación hay que explicar cómo funciona Salesforce “por debajo”. Hasta ahora parecía que todo iba sobre ruedas y que habíamos sorteado las piedras que nos íbamos encontrando por el camino a la hora de realizar este trabajo en cuanto a costes, fiabilidad o integridad de datos, pero ahora nos encontramos con otra que debemos sortear: “**FutureHandler**”. Debido a que Salesforce es un servicio en la nube y su principal característica es siempre estar disponible y ofrecer al usuario la mejor experiencia posible, Salesforce no puede permitirse el lujo de esperar por nosotros en una operación de base de datos. Cada vez que se registra una operación sobre un registro, esta desencadena una llamada DML sobre la base de datos que no puede ser parada o cancelada por un agente externo, por lo que Salesforce impide cualquier llamada a un servicio web durante el guardado o inserción de los registros a través de Trigger, para garantizar que ningún sistema externo pueda alterar el funcionamiento interno de Salesforce.

¿Entonces cómo es posible que estemos llamando a un servicio web desde un Trigger? Ahí entran en juego los “FutureHandler”. Mentí cuando dije que no se podía llamar a un servicio web desde un Trigger, sí que se puede llamar a un servicio web desde un Trigger, pero de manera asíncrona. Sería como dividir los hilos de ejecución de un programa. Cuando se produce una inserción y desde ella se llama a una función que invoca a un servicio web se genera un hilo de ejecución a futuro: Ese hilo de ejecución futuro son los “**FutureHandler**”. Cuando Salesforce “tenga tiempo” se invocará esa llamada, pero la operación DML sobre nuestros registros ya habrá terminado.

Es por ello por lo que no se puede mostrar al usuario en el momento de la inserción/modificación de un registro un mensaje de error que provenga de un sistema externo, por lo tanto, dificulta las opciones de depuración de errores.

Si ocurre un error en la llamada de algún servicio desde un FutureHandler, ya sea porque no está disponible el servicio o porque ha habido un error en la transacción, se genera un registro de error (LOG) con los mensajes devueltos por el sistema ERP para su posterior depuración.

Esto además tiene otra ventaja y es que permite que el usuario continúe con sus tareas, aunque haya errores él no ve, así se mantiene una disponibilidad de la herramienta dependiente únicamente de Salesforce y no de nuestro trabajo.

En la generación de un objeto LOG los datos que se almacenan son los siguientes:

- **Contacto:** Si el error ha sido en el servicio de contactos se referenciará el contacto que ha dado error en este campo
- **Cuenta:** Si el error ha sido en el servicio de contactos se referenciará el contacto que ha dado error en este campo
- **Error:** Error devuelto por el servicio web
- **Solucionado:** Check que se marca manualmente si el error de ser log ha sido solucionado
- **Usuario:** Usuario que invoco al servicio que genero el error.

### 3.6 Conclusiones

Gracias al modelo de datos utilizado, las herramientas estándar que Salesforce ponen a nuestra disposición y nuestro trabajo, hemos logrado realizar una sincronización con nuestro sistema ERP de manera eficaz y totalmente transparente para el usuario, utilizando seguridad en la transferencia de datos y contando con control de errores integrado.

**Todo el código de Salesforce está disponible en el siguiente enlace:**

<https://github.com/Mrmarthos/TFG/tree/master/Salesforce>

## 4 Pruebas y resultados

### 4.1 Introducción

En este capítulo se mostrarán las pruebas realizadas para comprobar el correcto funcionamiento de nuestra integración.

### 4.2 Pruebas Unitarias

Para comprobar el correcto funcionamiento de la integración de ambos sistemas se ha recreado un flujo completo de trabajo que un usuario realizaría en Salesforce. Desde Salesforce iremos viendo cómo se van modificando los registros según las diferentes operaciones realizadas y sus correspondiente llamadas API. Para ver los cambios de nuestro sistema ERP tendremos que consultar la base de datos de Azure, ya que no disponemos de interfaz de usuario para la visualización de registros.

### 4.3 Dataset

Los datos que se utilizaran para las pruebas son totalmente ficticios y generados arbitrariamente a propósito para este proyecto.

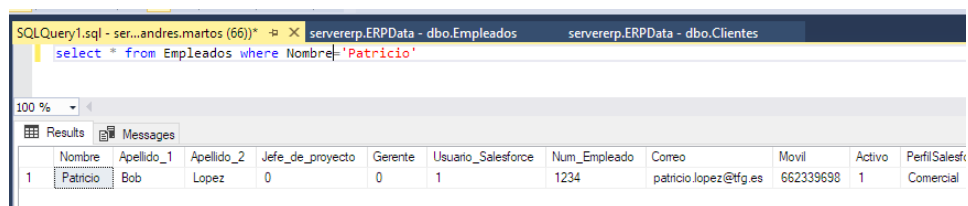
### 4.4 Resultados

#### 4.4.1 Sincronización empleados

En este apartado se detallarán las pruebas llevadas a cabo para la sincronización de los empleados, dividiéndola en dos apartados: Los empleados que tendrán usuario en Salesforce y los jefes de proyecto.

##### 4.4.1.1 Usuarios

Tenemos el siguiente empleado en la base de datos del sistema ERP:



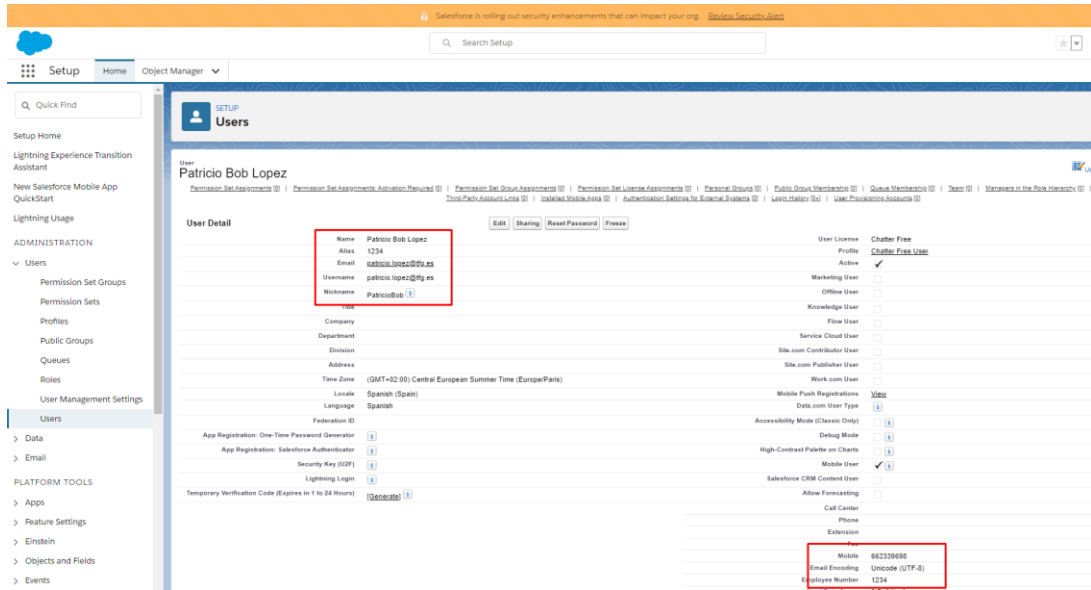
The screenshot shows a SQL query window with the following text: `select * from Empleados where Nombre='Patricio'`. Below the query, the results are displayed in a table with 11 columns: Nombre, Apellido\_1, Apellido\_2, Jefe\_de\_proyecto, Gerente, Usuario\_Salesforce, Num\_Empleado, Correo, Movil, Activo, and Perfil\_Salesforce. The table contains one row with the following values: Patricio, Bob, Lopez, 0, 0, 1, 1234, patricio.lopez@tfg.es, 662339698, 1, and Comercial.

	Nombre	Apellido_1	Apellido_2	Jefe_de_proyecto	Gerente	Usuario_Salesforce	Num_Empleado	Correo	Movil	Activo	Perfil_Salesforce
1	Patricio	Bob	Lopez	0	0	1	1234	patricio.lopez@tfg.es	662339698	1	Comercial

**Figura4.1:Consulta al sistema ERP sobre empleados**

Como vemos está marcado como usuario de Salesforce y está activo, por lo cual debe generar un usuario dentro de la plataforma

Si ejecutamos el batch y miramos nuestros usuarios de Salesforce vemos como se ha generado correctamente



**Figura4.2:Usuario generado en Salesforce desde sistema ERP**

\*El perfil no coincide debido a que no se disponen de más licencias por ser un entorno de pruebas

#### 4.4.1.2 Creación de jefes de proyecto

En la misma tabla vemos como hay registro creado marcado como gerente que debe generar un contacto:

SQLQuery1.sql - ser...andres.martos (66)\* X servererp.ERPData - dbo.Empleados servererp.ERPData - dbo.Cientes

```
select * from Empleados where Nombre='Jose'
```

100 %

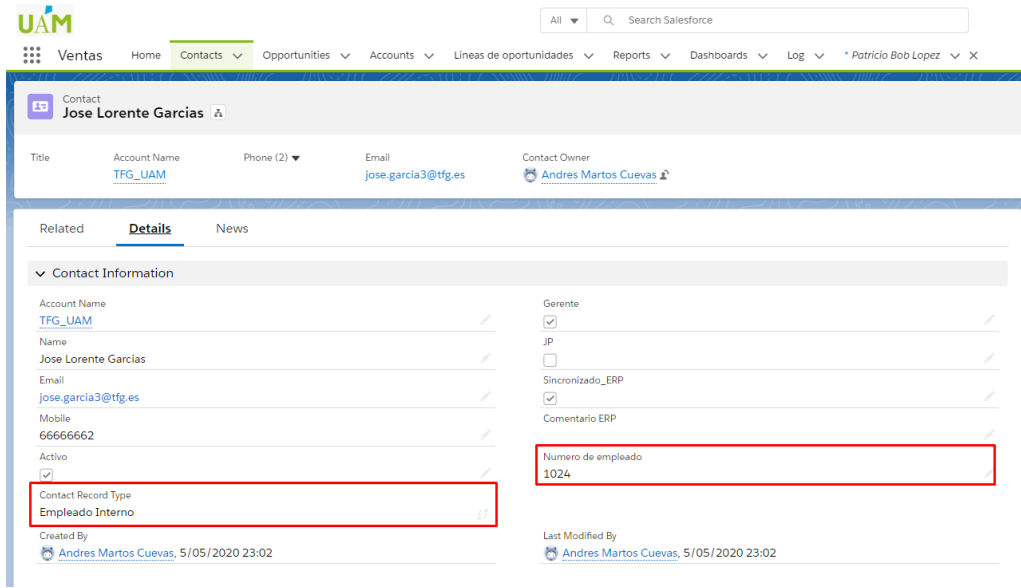
Results Messages

	Nombre	Apellido_1	Apellido_2	Jefe_de_proyecto	Gerente	Usuario_Salesforce	Num_Empleado	Comeo	Movil	Activo	PerfilSalesforce
1	Jose	Lorente	Garcias	0	1	0	1024	jose.garcia3@tfg.es	66666662	1	Comercial+

**Figura4.3:Consulta al sistema ERP sobre jefes de proyecto**



El mismo proceso batch ejecutado anteriormente ha generado también los registros de jefe de proyecto en Salesforce:



**Figura4.4:jefe de proyecto generado en Salesforce**

## 4.4.2 Sincronización de contactos

La sincronización de los contactos la he dividido en dos partes: La primera en la que el usuario busca manualmente en el componente Lightning contactos relacionados con una cuenta en este caso con la UAM. En la segunda parte se creará un registro directamente sobre Salesforce y deberá generar su correspondiente entrada en la tabla de contactos del sistema ERP.

### 4.4.2.1 Búsqueda manual de contactos

Un usuario de Salesforce buscaría los contactos relaciones con la empresa llamada UAM.

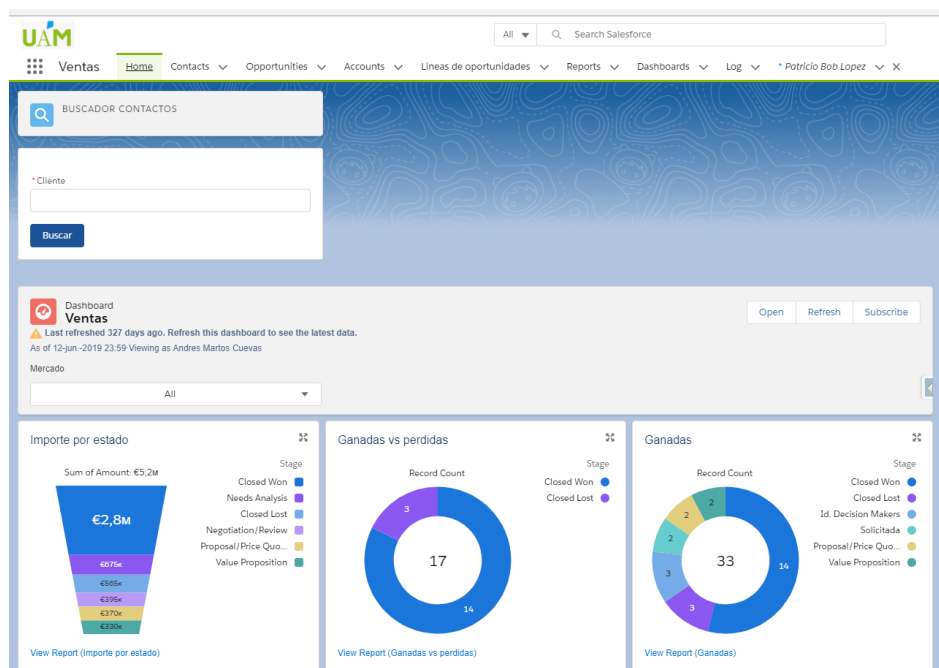
- Registros existentes en la base de datos del sistema ERP

```
SQLQuery2.sql - ser...andres.martos (64)* X servererp.ERPData - dbo.Contactos
select * from Contactos where Contactos.Cuenta_id IN ( Select Clientes.Id from Clientes where Clientes.Nombre LIKE 'UAM' )
```

	Correo	Movil	Telefono	Nombre	Apellido_1	Apellido_2	Cuenta_id	IdSalesforce	id	Comentario
1	Luisdepedro@uam.es	5465454	91854545	Luis	De	Pedro	41	NULL	4	Profesor TFG
2	Javierlopez@uam.es	662339874	NULL	Javier	Lopez	Sanchez	41	NULL	22	NULL
3	julian.sancho@uam.es	662265339	NULL	Julian	Sancho	Ruiz	41	NULL	23	NULL
4	Daniel.sordo@uam.es	6622913200	NULL	Daniel	Sordo	Fernandez	41	NULL	24	NULL
5	Eloy.alarcon@uam.es	6622913200	918452588	Eloy	Alarcon	Lopez	41	NULL	25	NULL

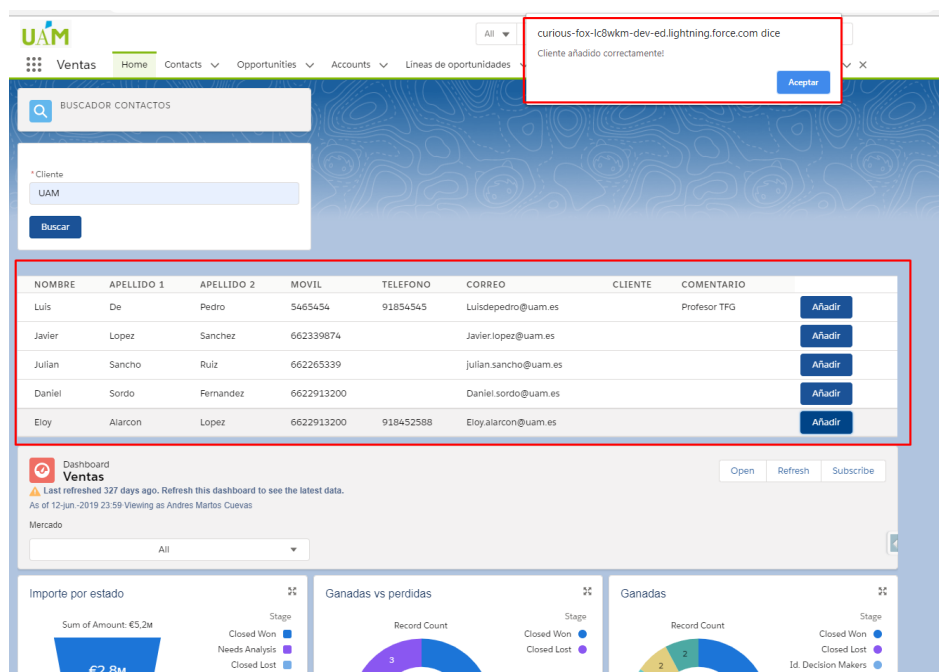
**Figura4.5:Consulta sobre contactos sistema ERP**

- Antes de realizar la búsqueda en Salesforce:



**Figura4.6:Buscador contactos Salesforce**

- Resultados que aparecen tras introducir UAM en el buscador:



**Figura4.7:Resultados buscador Salesforce**

De la lista de resultados, el usuario decide que quiere añadir a la plataforma el contacto “Eloy Alarcón”, salta un POP-UP, indicando que el contacto ha sido añadido correctamente, y si vamos a los contactos , el usuario ya tiene creado el contacto de “Eloy Alarcon” sin necesidad de introducir los datos manualmente.

The screenshot shows the Salesforce interface for a Contact record. At the top, the contact name is "Eloy Alarcon Lopez". Below this, there are fields for Title, Account Name, Phone (2) 918452588, Email eloy.alarcon@uam.es, and Contact Owner Andres Martos Cuevas. The main section is titled "Details" and contains a "Contact Information" section with fields for Account Name, Name (Eloy Alarcon Lopez), Email (eloy.alarcon@uam.es), Codigo ERP (25), Comentario ERP, and Gerente (checkbox). To the right of these fields, there are fields for Contact Owner (Andres Martos Cuevas), Phone (918452588), Mobile (6622913200), Last Modified By (Andres Martos Cuevas, 3/05/2020 19:35), and Created By (Andres Martos Cuevas, 3/05/2020 19:35).

**Figura4.8:Prueba sobre contacto en Salesforce**

#### 4.4.3 Sincronización PEPS y Cuentas

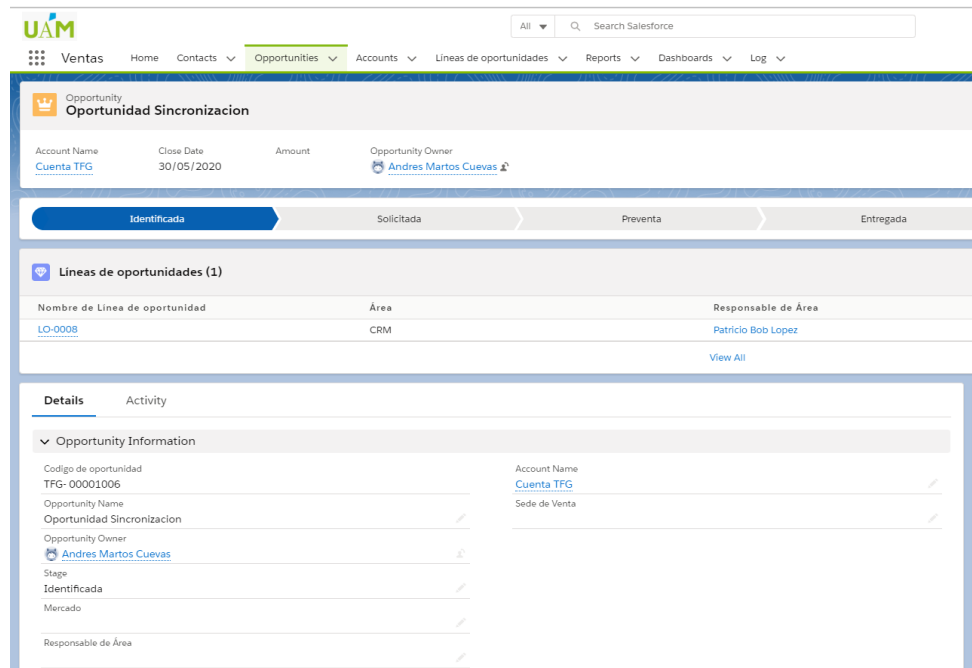
Para este escenario, el usuario creara una nueva cuenta en Salesforce, que no existe en el sistema ERP. Sobre esa cuenta se generará una oportunidad y sobre esa oportunidad se generará una nueva línea de oportunidad. Estos dos objetos (Cuenta, y línea de oportunidad) se deberán sincronizar automáticamente a medida que el usuario avanza en la oportunidad

- El cliente que el usuario ha creado en Salesforce es el siguiente:

The screenshot shows the Salesforce interface for an Account record. At the top, the account name is "Cuenta TFG". Below this, there are fields for Account Owner (Andres Martos Cuevas), Account Record Type (Cuenta Externa), and ID ERP. The main section is titled "Details" and contains an "Account Information" section with fields for Account Name (Cuenta TFG), Parent Account, NIF/CIF (58874525G), Razon Social (Cuenta TFG.SL), and Account Owner (Andres Martos Cuevas). To the right of these fields, there are fields for Sector and Shipping Address. Below the "Account Information" section, there is an "Address Information" section with fields for Billing Address (Campus UAM, 28013 Alcobendas Madrid, España) and Shipping Address. At the bottom, there is an "Additional Information" section with fields for Sincronizado\_ERP (checkbox), ID ERP, Created By (Andres Martos Cuevas, 9/05/2020 17:55), and Last Modified By (Andres Martos Cuevas, 9/05/2020 18:05).

**Figura4.9:Prueba sobre cliente en Salesforce**

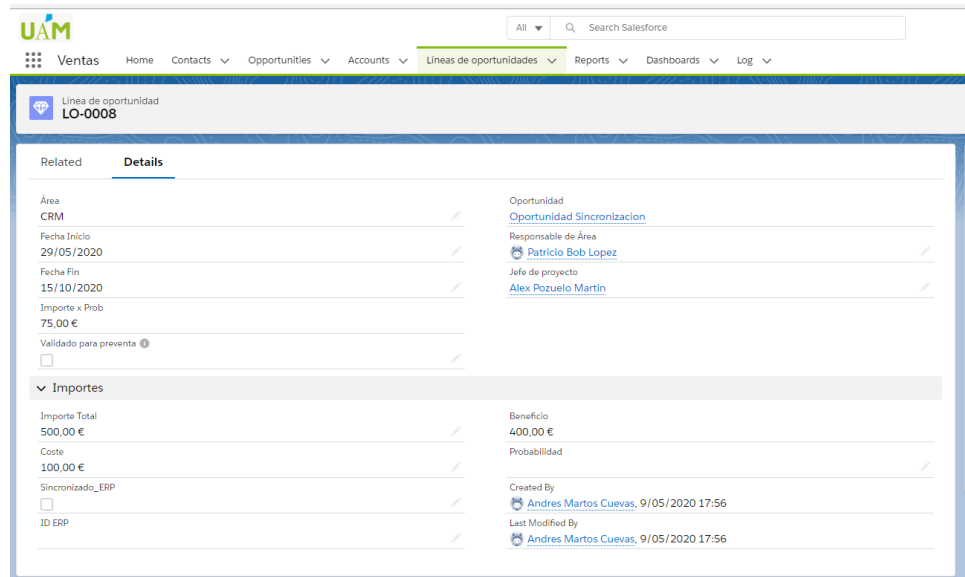
- Sobre ese cliente el usuario genera la siguiente oportunidad:



The screenshot shows the Salesforce interface for an Opportunity named "Oportunidad Sincronizacion". The top navigation bar includes "Ventas", "Home", "Contacts", "Opportunities", "Accounts", "Lineas de oportunidades", "Reports", "Dashboards", and "Log". The Opportunity details section shows the Account Name as "Cuenta TFG", Close Date as "30/05/2020", and Opportunity Owner as "Andres Martos Cuevas". Below this is a progress bar with stages: "Identificada" (active), "Solicitada", "Preventa", and "Entregada". A table titled "Lineas de oportunidades (1)" shows one line with "Nombre de Linea de oportunidad" as "LO-0008", "Área" as "CRM", and "Responsable de Área" as "Patricio Bob Lopez". The "Details" tab is selected, showing "Opportunity Information" with fields like "Codigo de oportunidad" (TFG-00001006), "Opportunity Name" (Oportunidad Sincronizacion), "Opportunity Owner" (Andres Martos Cuevas), "Stage" (Identificada), "Mercado", and "Responsable de Área".

**Figura4.9:Prueba sobre oportunidad en Salesforce**

- Sobre esa Oportunidad el usuario genera la siguiente “línea de oportunidad”

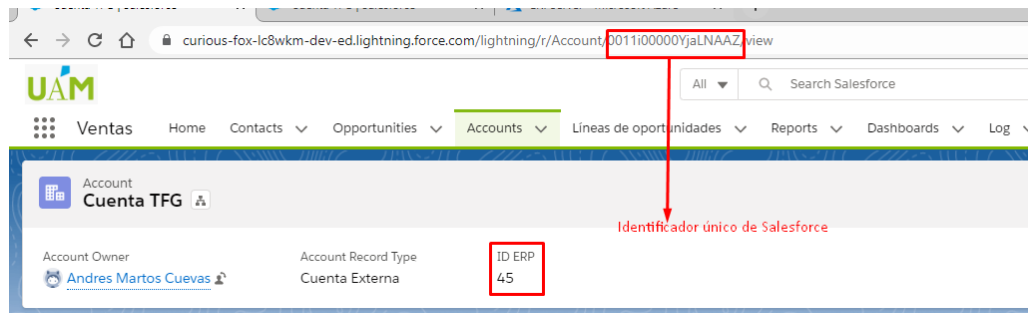


The screenshot shows the Salesforce interface for a specific Opportunity Line named "Linea de oportunidad LO-0008". The top navigation bar is similar to the previous screenshot, but the "Lineas de oportunidades" tab is selected. The "Details" tab is active, showing a table with fields like "Área" (CRM), "Fecha Inicio" (29/05/2020), "Fecha Fin" (15/10/2020), "Importe x Prob" (75,00 €), "Validado para preventa" (checkbox), "Importes" (Importe Total: 500,00 €, Coste: 100,00 €, Sincronizado\_ERP: checkbox, ID ERP), "Beneficio" (400,00 €), "Probabilidad", "Created By" (Andres Martos Cuevas, 9/05/2020 17:56), and "Last Modified By" (Andres Martos Cuevas, 9/05/2020 17:56).

**Figura4.10:Prueba sobre línea oportunidad en Salesforce**

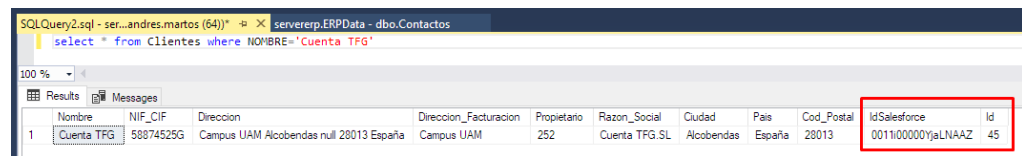
Una vez creado estos tres objetos en Salesforce el usuario avanza la oportunidad al estado “Solicitada” y cuando esto sucede, salta el Trigger que sincroniza la cuenta con el sistema ERP. Esto hace que se informe el campo “ID ERP” en la cuenta de Salesforce y que se genere la cuenta en la base de datos ERP:

- Salesforce:



**Figura4.11:Prueba sobre sincronizacion cliente Salesforce**

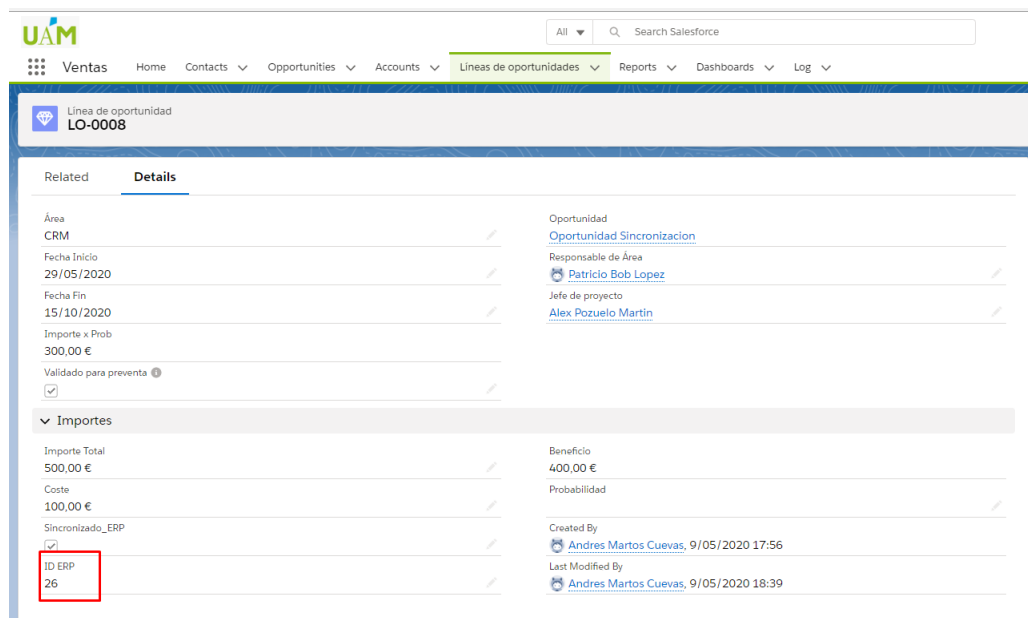
- Generación de cuenta en sistema ERP:



**Figura4.12:Prueba sobre sincronizacion cliente sistema ERP**

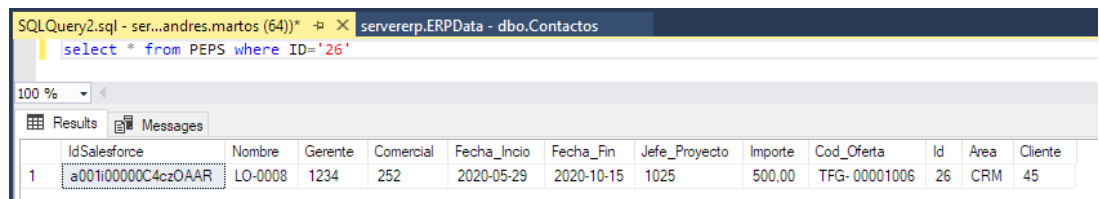
El usuario entonces decide pasar la oportunidad al estado preventa, lo que hace que se lance la sincronización de las líneas de Oportunidad/PEPS.

- En Salesforce, como ocurre con la cuenta, se marca el campo “ID ERP” con el valor que devuelve el servicio.



**Figura4.13:Prueba sobre sincronizacion línea de oportunidad Salesforce**

- En el sistema ERP se genera el PEP, con los datos provenientes de Salesforce: Empleados involucrados, fechas, importes...etc

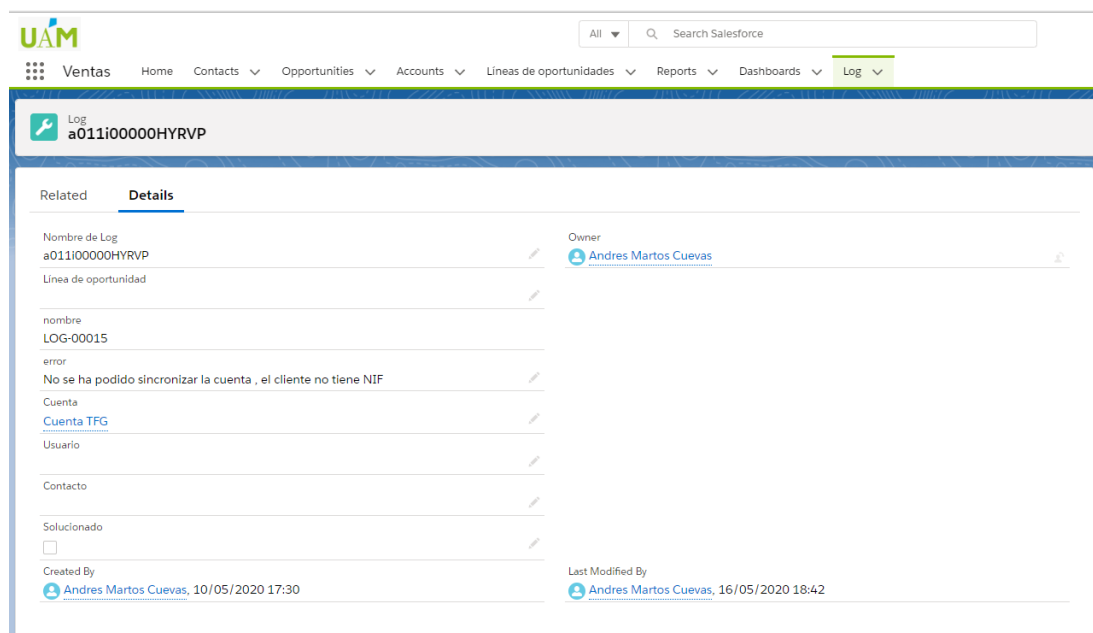


	IdSalesforce	Nombre	Gerente	Comercial	Fecha_Inicio	Fecha_Fin	Jefe_Proyecto	Importe	Cod_Oferta	Id	Area	Cliente
1	a001i00000C4czOAAAR	LO-0008	1234	252	2020-05-29	2020-10-15	1025	500.00	TFG- 00001006	26	CRM	45

**Figura4.14:Prueba sobre sincronizacion línea de oportunidad sistema ERP**

#### 4.4.4 Log de errores

Los registros de LOG tienen esta apariencia y en ellos se muestra información básica del error producido durante la sincronización, recogiendo el propio error devuelto por el sistema ERP



Related	Details
Nombre de Log	a011i00000HYRVP
Línea de oportunidad	
nombre	LOG-00015
error	No se ha podido sincronizar la cuenta , el cliente no tiene NIF
Cuenta	Cuenta TFG
Usuario	
Contacto	
Solucionado	<input type="checkbox"/>
Created By	Andres Martos Cuevas, 10/05/2020 17:30
Last Modified By	Andres Martos Cuevas, 16/05/2020 18:42

**Figura4.14:Ejemplo de log automáticamente generado en Salesforce**

## 4.5 conclusiones

La batería de pruebas realizada para cada una de las entidades confirma que la sincronizacion de ambos sistemas funciona correctamente, generando los registros y cambios en cada uno de los sistemas implicados.

# 5 Conclusiones y trabajo futuro

---

## 5.1 Conclusiones

Las conclusiones más relevantes obtenidas tras el desarrollo del presente trabajo de fin de grado se detallan a continuación. **El objetivo general que se estableció fue la creación de un sistema mediante el cual pudiéramos sincronizar dos herramientas muy diferentes.** Cada una de ellas con sus mecanismos y particularidades, pero que dependían la una de la otra ,y que son piezas claves en el correcto funcionamiento de cualquier empresa moderna hoy día. Tras su desarrollo se puede afirmar que:

- Es una implementación con **bajo coste económico** respecto a otras opciones. Gracias a la utilización de herramientas proporcionadas por ambos sistemas , no se hace necesario la utilización de plataformas externas o herramientas de pago ofrecidas por otras compañías. Se pueden consultar los datos económicos en el siguiente anexo: Q [Cálculos económicos](#)
- Es **flexible**. Durante su ciclo de vida, esta solución puede adaptarse a nuevas necesidades gracias a la flexibilidad inherente a los servicios WEB sin tener que realizar un nuevo desarrollo partiendo de cero.
- Es **mantenible**. Gracias a la creación de LOGS se pueden depurar los errores más fácilmente, limitando en número de personas necesarias para mantener en funcionamiento la solución. En el siguiente anexo se pueden ver como los errores han disminuido a lo largo del tiempo. R [Mantenibilidad](#)
- Es **escalable**. Una vez generada la estructura básica, de ella pueden crearse fácilmente nuevas ramas para la sincronización de otros objetos.
- **Transparente para el usuario**. Debido a los automatismos implementados, el usuario final no ve modificado su flujo de trabajo normal y no es necesario que realice tareas específicas para la sincronización de ambos sistemas.

Este sistema de sincronización está siendo utilizado actualmente en un entorno de producción real en una empresa internacional que cuenta con unos 1700 empleados. Hasta la fecha este mecanismo ha servido para sincronizar más de las 400 cuentas, 2300 PEPS y unos 400 contactos. Lleva puesto en producción aproximadamente desde principios de 2019 y los fallos ocurridos hasta la fecha han sido sobre todo debido a integridad de datos, no del propio funcionamiento de la solución.

Por todo esto podemos afirmar que hoy en día no encontraremos una solución más barata y fiable que resuelva la sincronización de Salesforce y un sistema ERP.

## 5.2 Trabajo futuro

Debida a que esta solución está siendo aplicada en el mundo real, han ido surgiendo ideas de mejora a medida que la solución era usada masivamente.

Una de la línea de desarrollo donde la solución puede mejorar está relacionada con la gestión de errores. Principalmente quedaría pendiente un desarrollo que permita enviar notificaciones por correo electrónico cada vez que se produjera un error de sincronización, haciendo que el tiempo de respuesta y de resolución de incidencias bajara drásticamente.

Además, sería también bueno realizar un desarrollo para que el usuario que ha provocado el error pudiera solucionarlo el mismo (siempre que sea de integridad de datos) si la necesidad de la intervención de un Administrador.

Otros de los puntos que quedan pendientes sería la realización de una encuesta de satisfacción a los usuarios para poder añadir nuevas funcionalidades propuestos por estos, mejorar los puntos peor valorados y subsanar errores que se nos hubieran podido pasar por alto

Por último, la URL del repositorio donde se encuentra el código fuente del sistema desarrollado es la siguiente:

<https://github.com/Mrmarthos/TFG>







# Referencias

---

- [1] Sistemas de planificación de recursos empresariales. [https://wikipedia.org/wiki/Sistema\\_de\\_planificaci3n\\_de\\_recursos\\_empresariales](https://wikipedia.org/wiki/Sistema_de_planificaci3n_de_recursos_empresariales) (Consultado el 15/01/ 2020)
- [2] Gotham K ,Create and publish ASP.NET WEB API In Azure. <https://www.c-sharpcorner.com/article/create-and-publish-asp-net-web-api-in-azure/> (Consultado el 15/01/ 2020)
- [3] Guía de introducción para desarrolladores de Azure. <https://docs.microsoft.com> (Consultado el 15/01/ 2020)
- [4] Guía para desarrolladores de Salesforce. <https://trailhead.salesforce.com/> (Consultado el 16/01/ 2020)
- [5] Jon Jensen, Build a REST Service in C#, <https://youtube.com> (Consultado el 20/01/ 2020)
- [6] Abey Francis, Enterprise Resource Planning, <https://www.mbaknol.com/management-information-systems/enterprise-resource-planning-erp-definition/>(Consultado el 20/01/ 2020)
- [7] Lightning Apps and Component, <https://lightningdesignsystem.com/platforms/lightning> (Consultado el 24/01/ 2019)
- [8] Precios de Servicios AWS ,<https://aws.amazon.com/es/pricing/> (Consultado el 16/01/ 2020)
- [9] Precios de Servicios Azure, <https://azure.microsoft.com/es-es/pricing>(Consultado el 16/01/ 2020)
- [10] Precios de Servicios Google, <https://cloud.google.com/pricing>(Consultado el 16/01/ 2020)
- [11] V. Kumar & Werner Reinartz. “Customer Relationship Management. Concept, Strategy, and Tools.” 1º Edici3n Springer , 2012
- [12] Sales Cloud, <https://www.salesforce.com/es/products/sales-cloud/features/> (Consultado el 18/01/ 2020)



## Glosario

---

- **API:** Application Programming Interface.
- **DML:** Data Manipulation Language.
- **CPD:** Centro procesamiento de datos.
- **CRM:** Customer Relationship Management.
- **REST:** Representational State Transfer.
- **SOAP:** Simple Object Access Protocol.
- **APEX:** Lenguaje de codificación de Salesforce.
- **Getter:** Función para obtener el valor asociado a una variable.
- **Setter:** Función para obtener el valor asociado a una variable.
- **Trailhead:** Organización Salesforce que puede utilizar para completar retos prácticos y probar nuevas funciones.
- **Developer Console:** Consola de programación de Salesforce.
- **Triggers:** Desencadenador de acción en operación de base de datos.
- **Batch:** Proceso de ejecución por lotes.
- **Query:** Consulta sobre base de datos.
- **AppExchange:** Mercado de aplicaciones para Salesforce
- **ERP:** Enterprise Resource Platform
- **Dataloader:** Herramienta para realizar cargas masivas en Salesforce

## Anexos

---

### A Modelo básico de datos de Salesforce

Salesforce se basa en un modelo de objetos. Dichos objetos están relacionados entre sí y pueden almacenar la información que el usuario quiera. Pero antes de guardar cualquier información es necesario conocer cuáles son estos objetos y que tipo de información deberían almacenar, después de todo la idea es tener organizada toda la información de clientes para usarla de forma simple.

Salesforce organiza los datos en registros, haciendo una analogía el objeto sería una hoja de Excel y el registro una fila de datos. Sin embargo, a diferencia de una hoja de Excel estos registros están relacionados con otros registros de otros objetos. Salesforce incluye una serie de objetos estándar con campos listos para ser usados por los usuarios sin necesidad de ninguna configuración previa. Estos objetos son:

- Cuentas:** Son las empresas o los clientes sobre el que estableceremos una relación comercial. En los campos estándar se puede almacenar información como la razón social, dirección, teléfonos, país...etc
- Contactos:** Son las personas relacionadas con una cuenta con las que vamos a realizar los negocios. Se guarda información de la cuenta, nombre, teléfonos...etc
- Lead:** Es una empresa o individuo del que queremos conocer si está interesada en alguno de los productos o servicios que vendemos.
- Oportunidades:** Uno de los objetos centrales de Salesforce, es un Lead convertido, que almacena información de la cuenta y el contacto y es aquí, en este objeto donde se almacenara toda la información relacionada con la gestión comercial.
- Campañas:** Es un objeto mediante el cual puede organizar una campaña de Marketing, agregar Contactos o Leads y hacer un seguimiento para determinar como de efectiva fue.

Dentro de los objetos estándar mencionados anteriormente Salesforce permite añadir tantos campos y relaciones como se desee. Del mismo modo se permiten crear objetos personalizados. Los objetos personalizados son unos de los grandes aciertos de Salesforce, ya que permiten cubrir las necesidades particulares de cualquier tipo de negocio.

Entrando más en detalles técnicos, las aplicaciones de Salesforce se desarrollan íntegramente en la nube y son almacenadas en la infraestructura de Salesforce.com. El lenguaje de programación utilizado para el desarrollo de estas aplicaciones es APEX, Javascript y Visualforce.

## B Coste

Salesforce cuenta con diferentes paquetes para adaptarse a las necesidades de las empresas.

- Salesforce Essentials:** Este paquete está enfocado para pequeñas empresas y tiene un coste aproximado de 25€ por usuario, siendo el máximo de usuarios permitidos 10. Tiene funcionalidades básicas como la gestión de cuentas y contactos y el seguimiento de oportunidades

- Lightning Professional:** Es una solución para empresas de cualquier tamaño. Su precio es de 75€ por usuario mes y añade funcionalidades como envío de correo electrónico masivo o gestión de campañas de marketing.

- Lightning Enterprise:** Esta es el paquete más vendido de Salesforce. Tiene un coste de 150€ mensuales por usuario y contiene casi todas las funcionalidades de Salesforce, pero con limitaciones. Estas limitaciones suelen estar muy por encima de las necesidades de la empresa media.

- Lightning Unlimited:** La versión más completa. Este paquete ofrece todas las funcionalidades disponibles sin límites. Tiene un coste de unos 300€ mensuales por usuario, por lo que está enfocada a empresas que vayan a hacer un uso muy intensivo de la plataforma.

Las versiones son incrementales, es decir, van añadiendo más funcionalidad de la incluida en los paquetes inferiores. Todos los productos por usuario requieren un contrato anual.

## C Campos Cliente sistema ERP

Las empresas necesitan guardar información de sus clientes ya sea para facturarles, proponer ofertas o simplemente establecer relaciones empresariales con ellos. Del mismo modo que el objeto empleado se puede dividir la información en tres categorías:

### 1. Campos de información comercial:

Nombre del campo	Tipo	Descripción
<b>Nombre</b>	Varchar (50)	Hace referencia al nombre del cliente, es un campo obligatorio
<b>NIF_CIF</b>	Varchar (50)	Guarda el NIF O CIF del cliente o empresa
<b>Direccion</b>	Varchar (50)	dirección del cliente
<b>Direccion_Facturacion</b>	Varchar (50)	dirección de facturación del cliente
<b>Razon_Social</b>	Varchar (50)	Razón social del cliente
<b>Ciudad</b>	Varchar (50)	Ciudad del cliente
<b>País</b>	Varchar (50)	País del cliente
<b>Cod_Postal</b>	Varchar (50)	Código postal

### 2. Campos referenciales

Nombre del campo	Tipo	Descripción
<b>Propietario</b>	Integer	Es una clave secundaria que hace referencia al campo <b>Num_Empleado</b> de la tabla empleados. Idéntica quien es el propietario de la cuenta a efectos de facturación, trazabilidad, gestiones comerciales..etc

### 3. Campos identificadores

Nombre del campo	Tipo	Descripción
<b>Id</b>	Integer	Es la clave primaria de la tabla. Este campo es de tipo entero auto-incremental, es decir comienza en 0 y va aumentando su número automáticamente cuando se inserta un registro nuevo. No puede contener duplicados
<b>IdSalesforce</b>	Varchar (50)	Campo que guardara el identificador de Salesforce es un campo único, es decir solo puede aparecer una vez

Tabla 1 : Campos cliente sistema ERP



## D Campos Empleado sistema ERP

### 1. Campos de información personal:

Nombre del campo	Tipo	Descripción
<b>Nombre</b>	Varchar (50)	Hace referencia al nombre del empleado, es un campo obligatorio
<b>Apellido_1</b>	Varchar (50)	Guardara el primer apellido y es obligatorio
<b>Apellido_2</b>	Varchar (50)	Guardara el segundo apellido y es obligatorio
<b>Correo</b>	Varchar (50)	Almacena el correo empresarial del empleado, acepta valores vacíos.
<b>Móvil</b>	Varchar (50)	Almacena el no de móvil de empresa del empleado, es de tipo “varchar” y acepta valores vacíos

### 2. Campos de control:

Nombre del campo	Tipo	Descripción
<b>Jefe_de_proyecto</b>	Boolean	Marca si un empleado es jefe de proyecto.
<b>Gerente</b>	Boolean	Marca si un empleado es gerente
<b>Activo</b>	Boolean	Marca si un empleado esta activo
<b>Usuario_Salesforce</b>	Boolean	Establece si un empleado debe tener cuenta dentro del entorno de Salesforce
<b>PerfilSalesforce</b>	Varchar (50)	Campo de control que establecerá el tipo de perfil que tendrá el empleado en Salesforce

### 3. Campos identificadores

Nombre del campo	Tipo	Descripción
<b>Num_Empleado</b>	Integer	Es la clave primaria de la tabla. Este campo es de tipo entero auto-incremental, es decir comienza en 0 y va aumentado su número automáticamente cuando se inserta un registro nuevo. No puede contener duplicados

Tabla 2: Campos empleado sistema ERP

## E Campos PEPS sistema ERP

### 1. Información del PEP

Nombre del campo	Tipo	Descripción
<b>Nombre</b>	Varchar (50)	Hace referencia al nombre del PEP, es un campo obligatorio
<b>Fecha_Inicio</b>	Date	Fecha de inicio del PEP
<b>Fecha_Fin</b>	Date	Fecha de fin del PEP
<b>Importe</b>	Money	Importe del PEP
<b>Razon_Social</b>	Varchar (50)	Razón social del cliente
<b>Area</b>	Varchar (50)	Área donde se realizará el PEP
<b>Cod_Oferta</b>	Varchar (50)	Código de la oportunidad a la que pertenece el PEP

### 2. Campos identificadores

Nombre del campo	Tipo	Descripción
<b>Id</b>	Integer	Es la clave primaria de la tabla. Este campo es de tipo entero auto-incremental, es decir comienza en 0 y va aumentado su número automáticamente cuando se inserta un registro nuevo. No puede contener duplicados
<b>IdSalesforce</b>	Varchar (50)	Campo que guardara el identificador de Salesforce es un campo único, es decir solo puede aparecer una vez

### 3. Campos referenciales

Nombre del campo	Tipo	Descripción
<b>Gerente</b>	Integer	Hace referencia a un numero de empleado de la tabla empleados. Establece quien es el gerente que ejecutara el PEP
<b>Comercial</b>	Integer	Hace referencia a un numero de empleado de la tabla empleados. Establece quien es el comercial que ha vendido el proyecto

<b>Jefe_Proyecto</b>	Integer	Hace referencia a un numero de empleado de la tabla empleados. Establece quien es el jefe de proyecto que ejecutara el PEP
<b>Cliente</b>	Integer	Hace referencia a un cliente o cuenta.Establece quien es la cuenta asociada a dicho PEP.

**Tabla 3: Campos PEPS sistema ERP**

## F Campos Contactos sistema ERP

### 1. Información de contactos

Nombre del campo	Tipo	Descripción
<b>Correo</b>	Varchar (50)	Correo electrónico del contacto. Es obligatorio
<b>Movil</b>	nvarchar	Numero de móvil de contacto
<b>Telefono</b>	nvarchar	Guardara el teléfono del contacto
<b>Nombre</b>	Varchar (50)	Guardara el nombre del contacto. Es un campo obligatorio
<b>Apellido_1</b>	Varchar (50)	Guardara el primer apellido del contacto
<b>Apellido_2</b>	Varchar (50)	Guardara el segundo apellido del contacto
<b>Comentario</b>	varchar(MAX)	Se utiliza para guardar cualquier anotación acerca del contacto

### 2. Campos identificadores

Nombre del campo	Tipo	Descripción
<b>Id</b>	Integer	Es la clave primaria de la tabla. Este campo es de tipo entero auto-incremental, es decir comienza en 0 y va aumentando su número automáticamente cuando se inserta un registro nuevo. No puede contener duplicados
<b>IdSalesforce</b>	Varchar (50)	Campo que guardara el identificador de Salesforce es un campo único, es decir solo puede aparecer una vez

### 3. Campos referenciales

Nombre del campo	Tipo	Descripción
<b>Gerente</b>	Integer	Hace referencia a un numero de empleado de la tabla empleados. Establece quien es el gerente que ejecutara el PEP
<b>Comercial</b>	Integer	Hace referencia a un numero de empleado de la tabla empleados. Establece quien es el comercial que ha vendido el proyecto
<b>Jefe_Proyecto</b>	Integer	Hace referencia a un numero de empleado de la tabla empleados. Establece quien es el jefe de proyecto que ejecutara el PEP

Tabla 4: Campos Contactos sistema ERP

## G Campos Cuenta Salesforce

### 1. Información de la cuenta

Nombre del campo	Custom	Tipo	Descripción
<b>Nombre de la cuenta</b>	No	Texto	Almacena el nombre de la cuenta
<b>NIF/CIF</b>	Sí	Texto	Identificación fiscal de la empresa
<b>Sector</b>	Sí	Multi-Picklist	Sector asociado a la cuenta
<b>Razón Social</b>	Sí	Texto	Razón Social de la cuenta
<b>Dirección</b>	No	Address	Dirección de la cuenta
<b>Dirección facturación</b>	No	Address	dirección facturación de la cuenta
<b>Sincronizado ERP</b>	Sí	Checkbox	Marca si la cuenta esta sincronizada con el sistema ERP

### 2. Campos identificadores

Nombre del campo	Custom	Tipo	Descripción
<b>ID</b>	No	Texto	Identificador de registro único generado por Salesforce automáticamente
<b>ID ERP</b>	Sí	TEXTO	Almacena el identificador del sistema ERP

### 3. Campos Referenciales

Nombre del campo	Custom	Tipo	Descripción
<b>Propietario</b>	No	Look-Up (Usuario)	Establece una relación de pertenencia entre un usuario y la cuenta

**Tabla 5: Campos cuenta Salesforce**

## H Campos Contactos Salesforce

### 1. Información del contacto

Nombre del campo	Custom	Tipo	Descripción
<b>Nombre</b>	No	Texto	Almacena el nombre del contacto
<b>Email</b>	No	Texto	Email asociado al contacto
<b>Móvil</b>	No	Texto	teléfono asociado al contacto
<b>Tipo de contacto</b>	Sí	Tipo Registro	Identifica si es un jefe de proyecto o un contacto externo.
<b>Activo</b>	Sí	Checkbox	Marca si un contacto esta activo o no
<b>JP</b>	Sí	Checkbox	Marca si un contacto es jefe de proyecto
<b>Sincronizado ERP</b>	Sí	Checkbox	Marca si el contacto esta sincronizado con el sistema ERP
<b>Numero Empleado</b>	Sí	Texto	Almacena el número de empleado
<b>Gerente</b>	Sí	Checkbox	Marca si el contacto es un gerente
<b>Comentario ERP</b>	Sí	Texto	Almacena información relacionada con el contacto del sistema ERP

### 2. Campos identificadores

Nombre del campo	Custom	Tipo	Descripción
<b>ID</b>	No	Texto	Identificador de registro único generado por Salesforce automáticamente
<b>ID ERP</b>	Sí	TEXTO	Almacena el identificador del sistema ERP

### 3. Campos Referenciales

Nombre del campo	Custom	Tipo	Descripción
<b>Propietario</b>	No	Look-Up(Usuario)	Establece una relación de pertenencia entre un usuario y la cuenta
<b>Nombre de la cuenta</b>	No	Master-Detail(Cuenta)	Establece una relación entre la cuenta y el contacto

**Tabla 6: Campos Contacto Salesforce**

## I Campos Oportunidad Salesforce

### 1. Información de la oportunidad

Nombre del campo	Custom	Tipo	Descripción
<b>Nombre</b>	No	Texto	Almacena el nombre de la oportunidad
<b>Etapas</b>	No	Texto	Etapas en la que se encuentra la oportunidad (“Identificada”, “Solicitada”, “Preventa”, “Entregada”, “Ganada”, “Perdida”)
<b>Sector</b>	Sí	Picklist	teléfono asociado al contacto
<b>Importe Total</b>	Sí	Formula Field Currency	Importe total de la oportunidad calculado a partir de la suma del importe de las líneas
<b>Importe Estimado</b>	Sí	Currency	Importe estimado de la oportunidad
<b>Beneficio</b>	Sí	Formula Field Currency	Calcula automáticamente el beneficio de la oportunidad
<b>Margen</b>	Sí	Formula Field Porcentaje	Calcula automáticamente el margen de beneficio en porcentaje de la oportunidad
<b>Fecha Propuesta</b>	Sí	Date	Fecha de la propuesta de la oportunidad
<b>Fecha Cierre</b>	Sí	Date	Fecha en la que se cierra la oportunidad
<b>Fecha de Inicio</b>	Sí	Formula Field Date	Fecha en la que empezaría el proyecto si se ganase. El dato proviene de la línea de oportunidad con menor fecha de inicio
<b>Probabilidad</b>	No	Porcentaje	Porcentaje de probabilidad de ganar la oportunidad
<b>Fecha Fin</b>	Sí	Formula Field Date	Fecha en la que empezaría el proyecto si se ganase. El dato proviene de la línea de oportunidad con mayor fecha de fin

## 2. Campos identificadores

Nombre del campo	Custom	Tipo	Descripción
<b>ID</b>	No	Texto	Identificador de registro único generado por Salesforce automáticamente
<b>Código de la oportunidad</b>	Sí	Auto-Number	Auto incremental que identifica a la oportunidad con el formato TFG-{0000}

## 3. Campos Referenciales

Nombre del campo	Custom	Tipo	Descripción
<b>Propietario</b>	No	Look-Up(Usuario)	Establece una relación de pertenencia entre un usuario y la oportunidad
<b>Nombre de la cuenta</b>	No	Master-Detail(Cuenta)	Establece una relación entre la cuenta y la oportunidad

**Tabla 7: Campos Oportunidad Salesforce**



## J Campos Línea Oportunidad Salesforce

### 1. Información línea de oportunidad

Nombre del campo	Custom	Tipo	Descripción
<b>Nombre</b>	Sí	Auto-number	Auto incremental que identifica a la oportunidad con el formato LO-{0000}
<b>Importe Total</b>	Sí	Currency	Importe total de la línea
<b>Importe x Probabilidad</b>	Sí	Formula Field Currency	Multiplica el importe total por la probabilidad de la oportunidad
<b>Área</b>	Sí	Picklist	Identifica el área al que pertenece la línea
<b>Beneficio</b>	Sí	Formula Field Currency	Calcula automáticamente el beneficio de la oportunidad
<b>Margen</b>	Sí	Formula Field Porcentaje	Calcula automáticamente el margen de beneficio en porcentaje de la oportunidad
<b>Fecha de Inicio</b>	Sí	Formula Field Date	Fecha en la que empezaría el proyecto si se ganase. El dato proviene de la línea de oportunidad con menor fecha de inicio
<b>Fecha Fin</b>	Sí	Formula Field Date	Fecha en la que empezaría el proyecto si se ganase. El dato proviene de la línea de oportunidad con mayor fecha de fin
<b>Sincronizado ERP</b>	Sí	Checkbox	Marca si la línea esta sincronizada con el sistema ERP

### 2. Campos identificadores

Nombre del campo	Custom	Tipo	Descripción
<b>ID</b>	No	Texto	Identificador de registro único generado por Salesforce automáticamente
<b>ID ERP</b>	Sí	Texto	Almacena el id del Sistema ERP

### 3. Campos Referenciales

Nombre del campo	Custom	Tipo	Descripción
<b>Propietario</b>	No	Look-Up(User)	Establece una relación de pertenencia entre un usuario y la oportunidad
<b>Nombre de la oportunidad</b>	Sí	Master-Detail(Oportunidad)	Establece una relación entre la línea y la oportunidad
<b>Responsable de Área</b>	Sí	Look-Up(Usuario)	Establece que usuario es responsable de esa línea
<b>Jefe de Proyecto</b>	Sí	Look-Up(Contacto)	Establece que persona es jefe de proyecto.

**Tabla 8: Campos línea oportunidad Salesforce**

## K Campos LOG Salesforce

### 1. Información del log

Nombre del campo	Custom	Tipo	Descripción
<b>Error</b>	Sí	Texto	Mensaje de error que ha generado el log
<b>Solucionado</b>	Sí	Checkbox	Marca si el error ha sido solucionado

### 2. Campos identificadores

Nombre del campo	Custom	Tipo	Descripción
<b>ID</b>	No	Texto	Identificador de registro único generado por Salesforce automáticamente
<b>Nombre</b>	Sí	Auto-Number	Auto incremental que identifica al log con el formato LOG-{0000}

### 3. Campos Referenciales

Nombre del campo	Custom	Tipo	Descripción
<b>Propietario</b>	No	Look-Up(User)	Establece una relación de pertenencia entre un usuario y la oportunidad
<b>Contacto</b>	Sí	Look-Up(Contacto)	Registro del contacto que ha generado el LOG
<b>Nombre de la cuenta</b>	Sí	Look-Up(Cuenta)	Registro de la cuenta que ha generado e LOG
<b>Línea de Oportunidad</b>	Sí	Look-Up(Línea de Oportunidad)	Registro de la línea de oportunidad que ha generado el LOG
<b>Usuario</b>	Sí	Look-Up(Usuario)	Usuario que ha generado el LOG

**Tabla 9: Campos LOG Salesforce**

## L Campos Usuario Salesforce

### 1. Información de usuario

Nombre del campo	Custom	Tipo	Descripción
<b>Error</b>	No	Texto	Nombre del usuario
<b>Email</b>	No	Texto	Email del usuario
<b>Numero Empleado</b>	No	Texto	Número empleado del usuario
<b>móvil</b>	No	Texto	Número de móvil del usuario

### 2. Campos identificadores

Nombre del campo	Custom	Tipo	Descripción
<b>ID</b>	No	Texto	Identificador de registro único generado por Salesforce automáticamente

**Tabla 10: Campos LOG Salesforce**

## M Ejemplo de respuesta JSON

```
{
  "ErrorCode": 0, → Código de error
  "Results": 2, →Numero de resultados devueltos
  "ErrorMessage": "OK", → Mensaje de error
  "id": 0, →Id de la transacción
  "Data": [ →Array que contiene los datos solicitados en la operación
    {
      "Correo": "testtfg@tfg.es",
      "Movil": "66226988",
      "Telefono": "998955",
      "Nombre": "Andres",
      "Apellido_1": "Martos",
      "Apellido_2": "Cuevas",
      "id": "1", →ID del sistema ERP
      "Cuenta_id": "40",
      "idSalesforce": "", →ID salesforce
      "Comentario": "Todo ok",
      "Company": null
    },
    {
      "Correo": "testtfg2@tfg.es",
      "Movil": "6226262264",
      "Telefono": "911221",
      "Nombre": "Jose",
      "Apellido_1": "Perez",
      "Apellido_2": "Prieto",
      "id": "2",
      "Cuenta_id": "40",
      "idSalesforce": "",
      "Comentario": "Responsable Banca",
      "Company": null
    }
  ]
}
```

## N Ejemplo de respuesta JSON

```
{
  "ErrorCode":0, →Código de error(0 si todo ok)
  "Results":0,
  "ErrorMessage":"OK", →Mensaje de error
  "id":25, → Id del objeto en el sistema ERP
  "Data":null
}
```

## **O Tipos de nube**

### **1. Pública**

La nube pública se define como servicios informáticos que ofrecen proveedores externos a través de la Internet pública y que están disponibles para todo aquel que desee utilizarlos o comprarlos. Estos pueden ser gratuitos o venderse a petición lo, que permite a los clientes pagar solo por el uso que hacen de ciclos de CPU, el almacenamiento o el ancho de banda que consumen.

A diferencia de la nube privada, de la que hablaremos más adelante, las nubes públicas permiten ahorrar los gastos que suponen comprar, administrar y mantener el hardware e infraestructura de las aplicaciones locales. Las nubes públicas también se pueden implementar con más rapidez que las infraestructuras locales y con una plataforma que permite una escalabilidad casi ilimitada. Esto permite a los usuarios acceder a una aplicación o servicio desde cualquier lugar con el dispositivo que prefieran siempre y cuando tengan acceso a internet.

### **2. Privada**

La nube privada se define como los servicios informáticos que se ofrecen a través de Internet o de una red interna privada solo a algunos usuarios y no al público general.

También denominada nube interna o corporativa, la informática en nube privada aporta a las empresas gran parte de las ventajas de la nube pública (como autoservicio, escalabilidad y elasticidad), pero con el control y la personalización disponibles en los recursos dedicados a través de una infraestructura informática hospedada en el entorno local. Es la mejor opción para las compañías que necesitan la privacidad y protección de los datos, ya que ofrecen un nivel más alto de seguridad y privacidad con firewalls propios y hospedaje interno.

Un inconveniente es que el propio departamento IT de la empresa que posee la nube es el encargado de mantener y administrar los servicios por lo que suponen un mayor coste a nivel de personal, administración y mantenimiento.

### **3. Híbrida**

Las nubes híbridas combinan los modelos de nubes públicas y privadas. Un usuario es propietario de unas partes y comparte otras. Cuando la demanda computacional fluctúa, este tipo de nube brinda a las empresas la capacidad de escalar su infraestructura local hasta la nube pública, sin que los centros de datos de terceros puedan acceder a la totalidad de los datos.

Las organizaciones obtienen la flexibilidad y el poder informático de la nube pública para tareas informáticas básicas y no sensibles, al tiempo que mantienen las aplicaciones y los datos críticos para la empresa en las instalaciones, de forma segura detrás de un firewall de la empresa. Este tipo de nube combina lo mejor de los dos mundos, la flexibilidad, escalabilidad y rentabilidad de las nubes públicas y la protección de datos de las nubes privadas.

## P Servicios Ofrecidos

### 1. Infraestructure as a Service (IaaS)

En este caso se contrata capacidad de proceso (CPU) y almacenamiento. En este entorno se puede desplegar aplicaciones propias que por motivos de coste o falta de conocimientos no queremos instalar en nuestra propia empresa. El proveedor se encarga de su gestión y para el cliente se convierten todos los gastos en variables (sólo se paga por lo que se usa).

### 2. Platform as a Service (PasS)

Es una oferta de cloud computing que proporciona a los usuarios un entorno de cloud en el que pueden desarrollar, gestionar y distribuir aplicaciones. Además del almacenamiento y otros recursos informáticos, los usuarios pueden utilizar un conjunto de herramientas prediseñadas para desarrollar, personalizar y probar sus propias aplicaciones. Normalmente hay que seguir una serie de restricciones para poder desarrollar estas para un proveedor (por ejemplo, en cuanto a los lenguajes de programación).

### 3. Software as a Service (SaaS)

SaaS, o Software as a Service, es una forma de poner en disposición softwares y soluciones de tecnología por medio de la internet, como un servicio. Con ese modelo, la empresa no necesita instalar, mantener y actualizar hardware y softwares. El acceso es fácil y simples: solo es necesario conexión con la internet.

Las aplicaciones SaaS también son llamadas de softwares basados en Web, softwares on demand o softwares hospedados. Estos softwares son ejecutados en las empresas proveedoras, que tienen la responsabilidad de gestionar el acceso y mantener la estructura de datos, la conectividad y los servidores necesarios para el servicio.



Figura P-1: Esquema de arquitectura cloud

## Q Cálculos económicos

Realizando una estimación de costes para este proyecto y comparándolo con herramientas que ofrecen una funcionalidad similar hoy día, este desarrollo es rentable a partir de los **16 meses**. Esta afirmación se basa en los siguientes datos:

### 1. Costes derivados de la realización de los desarrollos:

Para la estimación de costes de este proyecto se ha tenido en cuenta únicamente los recursos humanos y centrados únicamente en la parte de Salesforce, debido a que no es necesario ningún material, software o licencia adicional a las ya adquiridas previamente.

- Estimación Jornadas : Para realizar todo el desarrollo de Salesforce, incluyendo pruebas y análisis se ha estimado una duración de **20 jornadas**.
- Costes en personal: Este proyecto puede realizarlo una sola persona con experiencia de unos 2 años trabajando en Salesforce cobrando un sueldo de **35.000€** anuales. Si aplicamos un coste estándar para la empresa de 1,5 el salario ,los cálculos serían:
  - Coste empresa  $35.000€ \times 1,5 = 52.500€$
  - Precio por jornada =  $52.500€ / 220(\text{jornadas año}) = 238.6€$

**Coste total proyecto= 4.772€**

### 2. Costes herramienta de pago.

En la tienda de aplicaciones de Salesforce podemos encontrar una aplicación de pago muy similar a la que se ha desarrollado en este TFG con un coste de 300\$ mensuales:

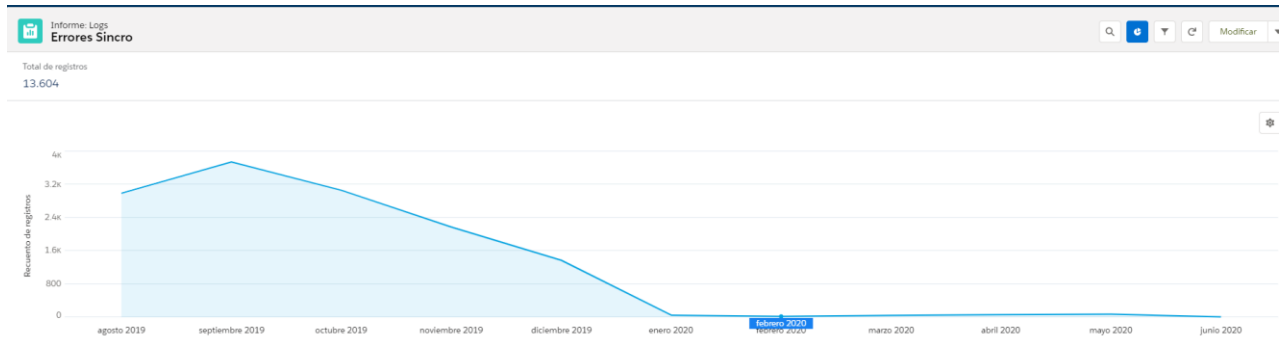
The screenshot displays the Salesforce AppExchange page for the 'SAP Enterprise Mashups' application. The header includes the AppExchange logo and a search bar. The main content area features a large image of the application interface, which shows a 'Universal Container - Enter 19001' screen. To the right of the image, there is a pricing section stating 'Starting at \$299 USD per company per month' with a 'Get It Now' button. Below this, there is a rating section showing a 5-star rating (5 reviews) and a 'LISTED ON 8/29/2006' date. The 'CATEGORIES' section lists 'Enterprise Resource Planning'. A 'Watch Demo' button is also present. The description text states: 'SGC Enterprise Mashups provide real-time SAP Integration. Leverage your Salesforce investment by providing your field teams real-time visibility to SAP sales or service orders, shipping status, credit status, pricing, availability and more. Even create SAP quotes and orders within Salesforce.com screens!'. A red box highlights the 'SAP Integration' text in the description. At the bottom of the image, there is a '1 of 4' indicator.

**Figura Q-1: Herramienta pago Salesforce -SAP**



## R Mantenibilidad

Gracias a los logs podemos ver como los errores van disminuyendo a medida que la sincronización se va utilizando y se van depurando los errores.



**Figura R-1: Grafica de LOGS**

Ese gráfico muestra la cantidad de logs generados en un entorno de producción desde agosto 2019(desde que se implanto) hasta la junio 2020.

